PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 12/54

A2

(11) International Publication Number: WO 98/24208

(43) International Publication Date: 4 June 1998 (04.06.98)

(21) International Application Number: PCT/GB97/03212

(22) International Filing Date: 24 November 1997 (24.11.97)

(30) Priority Data:
9624419.9 23 November 1996 (23.11.96) GB

(71) Applicant *(for all designated States except US)*: INMEDIA INVESTMENT LIMITED [GB/GB]; 155 Gower Street, London WC1E 6BT (GB).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: MUIRHEAD, Charles [GB/GB]; Flat 3, 141 Beaufort Street, London SW10 (GB). DONKIN, Richard [GB/GB]; Flat 3, 24 Keswick Road, London SW15 2JP (GB). JONES, Richard [GB/GB]; 170 Highbury Hill, London N5 1AU (GB). LAURIE, Ben [GB/GB]; 17 Perryn Road, London W3 7LR (GB). LAVELLE, Neil [GB/GB]; 58 Lorrimore Square, Kennington, London SE17 3QU (GB). ZANCANI, Leo [GB/GB]; 58 Lorrimore Square, Kennington, London SE17 3QU (GB).

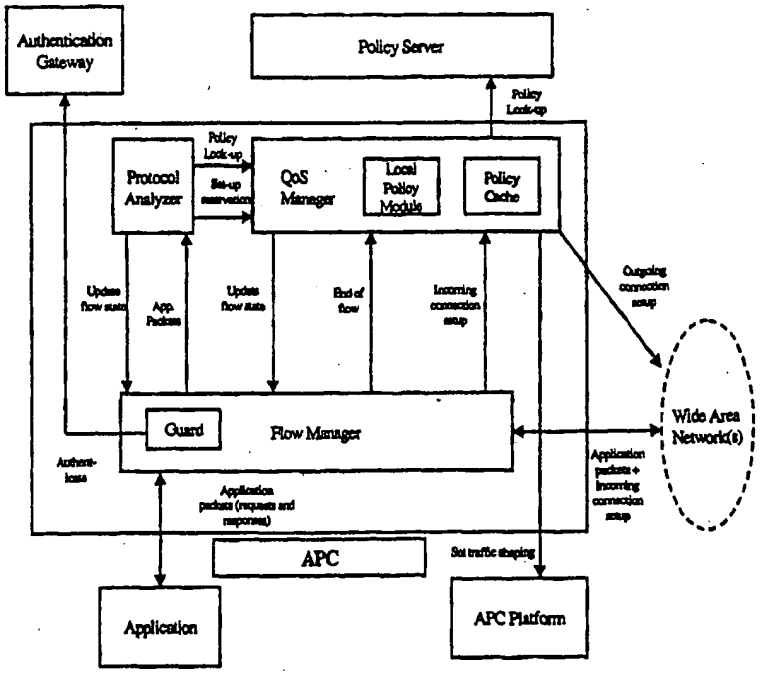
(74) Agent: LOVELESS, Ian, Mark; Reddie & Grose, 16 Theobalds Road, London WC1X 8PL (GB).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published

Without international search report and to be republished upon receipt of that report.

(54) Title: DATA COMMUNICATION SYSTEM



(57) Abstract

Methods and apparatus are provided for analysing data requests in a computer network and providing data with particular transfer parameters in accordance with the request, such as bandwidth, latency and jitter. Various ways of determining the appropriate transfer parameters include, analysing the node or user issuing the request, analysing the node serving the response, or analysing the request or response.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

DATA COMMUNICATION SYSTEM

Field of Interest

The present invention relates to computer communications and networking, and in particular the delivery of data such as advertising and electronic published content over electronic networks. More specifically, the present invention relates to the delivery of electronic data with a specified quality, and the delivery of advertising with specified quality and targeting.

Background of the Invention

A computer is a data processing system. A computer includes input/output devices (such as display terminals, keyboards, printers, long-term storage devices and the like) and some data processor(s) which supports the operation of these devices.

A computer typically runs an operating system which abstracts the technical details of the computer's input/output devices and provides useful common functionality and services to programs running on the computer.

A computer network is a set of data processing systems connected in such a way as to allow program units on different computers to exchange data. Each computer in a computer network is generically termed a node. Nodes are connected in a variety of ways by links from one node to another. Links pass data over a variety of electrical or optical media.

Data may pass over several links and through several nodes in the process of travelling from one program unit to another.

Fig. 1 Shows a schematic of a simple computer network. Circles 10 represent nodes and solid lines 20 represent links. The dashed line 30 represents an exchange of data between two particular nodes 11,12 which passes through two nodes in between.

Each node in a computer network may contain one or more program units. These program units have associated with them an address which is unique to the network. The effect is that program units may refer to each other for the purposes of exchanging data by using this unique address.

Application program units are application computer programs, whereas system program units are computer programs that are not related to any specific computer program, but provide generic services, typically as part of an operating system or communications protocol software. Some program units interact with human users in order to allow these human users to interact with the network. Application program units are not at the level of communications protocol software, but run at a level above; the protocol software providing services to the application program units.

Many (but not all) types of data exchange between program units take the form of client/server or request-reply transactions. In this model, a program unit known as the client sends a request for information to a program unit called the server which is known to have this information. Upon receiving this request, the server sends the requested information in a reply.

Application program units may be both clients and servers, that is they may both generate and serve requests.

A "proxy" is a system program unit that simply forwards a request from a client on to a server, and forwards the response from the server back to the client.

The cost (in terms of time and money) of accessing a distant server over a computer network is often a significant factor, and to reduce this cost, large banks of storage called proxy caches are placed between the client and the server. A proxy cache is a system program unit that acts as a proxy for a group of clients, but instead of merely forwarding requests and responses, stores up certain responses and serves them directly from a local store without consulting a server if the proxy has seen the request previously. The effect of a proxy cache is generally to reduce the cost of a client accessing a server.

Fig. 2 shows a schematic of part of a computer network employing a proxy cache. The local group of four client nodes 10 is connected to a proxy cache 40. The proxy cache itself is connected over a costly link shown as a dotted line 21 (it may pass through many other nodes, or may be a costly link to use, or may be a slow link) to a server 50.

The term "proxy cache" is often used colloquially to refer to a specific embodiment of a general proxy cache, namely one that caches only WWW requests made using the HTTP protocol over the Internet protocol. In this patent document, the term "proxy cache" will not be used in this way. This term, whenever it is used, will refer to the generic scheme outlined in the previous two paragraphs. The proxy cache is thus not limited to the HTTP IP protocol.

The term network service means one or more computer systems that are connected to a network and offers one or more data processing services to more than one node on that network. A network service is normally available by using a so-called "well-known address", the intent being that nodes on the network that wish to make use of the said network service can easily determine the address of this network service.

The Internet refers to a particular world-wide computer network. In the terminology of the Internet, nodes are called hosts and application program units are typically referred to as programs, clients or servers.

There are many de facto and de jure standards that have been built up over a period of some 30 years which have allowed standard computers to be turned easily into Internet hosts. These standards are collectively referred to as the Internet protocol suite, or just the Internet Protocol (IP). The IP suite is also sometimes referred to as TCP/IP because of the popular TCP transport protocol that runs over IP. A good summary of IP can be found in "TCP/IP Illustrated, Vol. 1, The Protocols" by W. Richard Stevens, published by Addison-Wesley, which is incorporated herein by reference.

IP defines a way to uniquely address application program units running on Internet hosts. Application program units are addressed by the hostname of the host on which they are running and by a unique port number which identifies the service provided on that particular host. It is possible for a single application program unit running on an Internet host to be identified by zero, one or more than one port number. This is usually done for programs that wish to offer more than just a single service or do not wish to offer any service at all. The tuple (hostname, port number), being an address, is sometimes referred to as a socket. An exchange of data between two Internet hosts has two endpoints, and therefore can be identified by the end sockets, known as a socket pair.

Since IP is just a standard way of connecting a group of computers together, there are computer networks which use IP but are not connected to the Internet, or are connected to the Internet only in some very limited way. These privately managed, computer networks are usually referred to as intranets. Although the background of the invention, and the preferred embodiments are described with respect to the Internet, the scope of the invention is not limited to the Internet or to intranets, nor is its scope limited to computer networks using the Internet protocol.

Traditional computer networks, in particular the Internet, have offered enormous amounts of published content that readers equipped with special software clients (browsers) can peruse. The Internet, for example, has been able to build up this content so quickly by making it easy for individuals and small and large companies to start publishing at very low costs. However, these networks have generally been slow, and suitable only for distributing text and small pictures. While it is possible to send multimedia (sound and video) content over the Internet, it is not done very frequently, and the quality of Internet multimedia services is normally very poor.

Sound and video streams require a high "service level" which includes high bandwidth, low latency, real time constraints, and other parameters which affect data delivery over a network. Such constraints are known as service level guarantees. A network which can offer service level guarantees is known as a quality of service (QoS) network. A service level is defined by the parameters which affect data flow across a network, such as bandwidth, latency, jitter and burstiness. Other parameters may also affect the service level. Bandwidth is the maximum rate of data flow through any particular link in a network. Latency is the delay between a message leaving one node and arriving at another. In a network without a traffic load the latency will simply be the physical transfer time between two nodes. In a loaded network, queuing delays (caused by a backlog of messages waiting for transmission) reduce the rate of message transfer and thus increase the latency. Jitter is the variation in latency from one message to another, typically due to varying queuing delays. Burstiness is the degree to which the bandwidth required by the application program unit varies over time.

A high level of service has one or more of high bandwidth, low latency and low jitter. A network having provision to supply at a particular guaranteed service level is a QoS network. Such networks frequently make use of the burstiness parameter, since this enables more efficient use of bandwidth through statistical multiplexing. Other parameters than

bandwidth, latency, jitter and burstiness may be taken into account in a QoS network. Examples of QoS network protocols are Resource Reservation Protocol (RSVP) and Asynchronous Transfer Mode (ATM) protocols.

The Internet itself, because of its distributed ownership and lack of pricing structure, does not at present offer QoS guarantees. For example, Internet message packets transmitted by one host to another may never reach the destination host or may even appear at the destination host more than once, and there is in general no limit on the amount of time that the Internet may take to transmit data from one host to another. This type of network is usually known as a best effort network. Current schemes which propose to extend the Internet infrastructure to add these QoS guarantees either:

require extensive changes not just to the network itself, but to the software (applications and IP implementations) that people use on their own machines (with upwards of 30 million people currently estimated to be using the Internet, this cost is very significant). Examples of such network and software changes are those required to handle networks supporting the Resource Reservation Protocol (RSVP) or Asynchronous Transfer Mode (ATM). Changes to application program units to enable them to be aware of QoS are undesirable as this requires linking the application with low level details of the computer network and also results in the application becoming dedicated to that network only; or

offer QoS guarantees using only the host name or port number or socket pair (many requests may go through a single port number or socket pair). This simply attempts to provide the same QoS for all data transfer between the specified pair. Since many requests, each potentially requiring a different QoS, can be sent with the same port number or socket pair, selecting the correct QoS is only possible if the maximum conceivable QoS is supplied to all requests.

For example, since requests to transfer large video files may use the same HTTP socket pair as requests for small text files, the QoS for video files must be assigned to all requests on that socket pair, simply to ensure that video files are delivered correctly - this is, of course, extremely wasteful of network resources. HTTP is not the only protocol that enables very diverse requests to be sent over a single socket pair - others include: DCOM (Distributed Component Object Model) and CORBA IIOP (Common Object Request Broker Architecture, Internet Inter-ORB Protocol), which are increasingly popular for complex applications; database access protocols such as Oracle's SQL*Net, which can be used for short transactions as well as returning very large volumes of data; and many other application-specific protocols.

There exist other network technologies at present that allow QoS guarantees to be made and allow the network to decide whether it can or cannot meet those guarantees. One example is the telephone network. When a number is dialled, the telephone network decides whether there are spare lines available from end to end, and then reserves those lines for the duration of the call. If lines are not available at any point along the route, it returns a "service not available" tone. Thus the caller makes a contract with the telephone network that it will provide the necessary bandwidth and latency for a voice conversation for the duration of the call (and the caller, of course, pays for this privilege). There are computer network technologies which work in a similar manner (though they are usually much more powerful than this, for instance allowing the caller to set exact minimum bandwidth requirements). We call these QoS networks.

Some networks can offer either a QoS service or a best effort service. A best effort service does not provide service level considerations, and so is not considered to be a QoS service level. Internet Protocol is an example of a best effort protocol in which packets are transmitted and can arrive at their destination with unpredictable delays. The best

effort IP network tries to ensure that a certain proportion of packets arrive at their destination, but makes absolutely no guarantees and in times of congestion may completely fail to deliver some users' packets. Transmission Control Protocol (TCP) assists this by re-transmitting packets which are not received. However, neither TCP or IP can provide a specified level of service.

Current best effort networks are very difficult to bill accurately. Indeed, the largest best effort network available now - the Internet - has no effective means of billing save for a simple monthly flat rate agreement The problems that arise with billing of best effort services are two-fold: first, users are unwilling to enter into a financial contract with a service that makes no guarantees about when or even whether data will be delivered; secondly, the packet oriented rather than connection-oriented service which usually characterises best effort networks defeats traditional billing systems that normally compute the duration and distance of "calls". This has several implications which, in fact, have been amply demonstrated on the Internet: users are not penalised for abusing the service by requesting enormous data transactions and there is little economic incentive to develop protocols which use network bandwidth efficiently.

In contrast, QoS networks work in a connection-oriented manner, with a connection being administered by a central or distributed network mechanism, with the connection staying 'up' for a measurable amount of time, and with some particular QoS parameters being associated with the connection while it is up. Billing for transactions on QoS networks is therefore potentially much simpler than in best effort networks. Nevertheless, the billing systems used by telecommunication companies still have difficulty handling the QoS parameters, and at present there is no agreement within the industry about a sensible way to bill for such connections. Furthermore, it is at present still very difficult to identify the content being transmitted over a connection and charge an

"all-inclusive" fee that covers both the cost of transmitting the information and the cost of the information itself.

Proxy caches potentially reduce the number of requests and replies that need to be transmitted over expensive wide area network links. However, servers often attempt to subvert caches using two strategies: either they mark the data as "not cacheable" using a flag that forces the Proxy cache to ignore it for caching purposes, or else they mark the data with a very short expiry date, forcing the Proxy cache to reread the data more frequently than it would otherwise have done. Servers attempt this subversion in order to collect content usage counts for particular items of information. This content usage count information is usually collected in order to produce accurate viewing figures for advertisers, or merely to measure or prove how popular a particular piece of information is. This practice reduces the effectiveness of Proxy caching markedly.

Users often do not have a subscription to any particular content provider's site, preventing such content sites from collecting demographic information about - for example - the class of people visiting a certain piece of information. Certain client-server protocols allow such information to be collected. The way this is done is currently rather clumsy. For example the popular HTTP protocol (which relies on TCP and IP) allows so-called "cookies" to be issued to clients. These cookies are usually issued after the user has provided some information about themselves, and allow the user access to a particular server. Thus, access to a server effectively becomes conditional upon the user explicitly providing demographic information to the server.

Traditional media offer a wide spectrum of services from broadcast, high quality delivery (e.g. TV, cinema, radio) to targeted low quality delivery (e.g. direct mail, advertising in specialist magazines). There are various media which fall between these two extremes, e.g. newspaper advertising and billboards. Fig. 3 summarises the current media

alternatives. No traditional or new media currently offer high quality, highly targeted advertising to a mass audience. This is a "holy grail" of advertising. At the same time, a variety of credit card companies and supermarkets do off-line processing of sales transactions in order to provide accurate targeting information for advertisers. Unfortunately, this off-line processing of sales receipts has limited scope. It prevents advertisers from, for example, targeting customers who may be interested in a particular product but may not have previously purchased that product or a similar type of product. One example might be in the targeting of advertising for a colour TV.

Since people buy colour TVs relatively infrequently, it is difficult to target a potential customer simply using past sales receipts.

The Internet is a rather different medium from traditional advertising channels. The Internet currently offers a slow, low quality, `best effort' service. There is no method at present of paying to ensure that certain traffic is given a higher priority of service. In other words, all Internet traffic is treated equally.

This has various implications for advertisers. The poor-quality, best effort, overused Internet backbone prevents TV-quality advertising from being delivered: only low quality pages of text and small images can be delivered in a reasonable time. Unlike TV, advertisers cannot pay to deliver adverts with guaranteed quality of service QoS - e.g. with streaming media quality and responsiveness.

Using the Internet, advertisers can provide a great deal more product information which allows users to make more informed purchasing decisions. Far more information can be conveyed by a Web advert than by a 30 second TV advert. Advertisers can provide links to more detailed information and specifications which readers can follow up immediately if they are interested - essentially because the Internet is an interactive, not a broadcast medium.

There are, extremely good emergent technologies for dynamically targeting advertising over the Internet. This targeting tracks users as they actually read published content on the Internet, thus complementing traditional targeting methods mentioned above. It should be possible to target potential customers based on their reading preferences, not just on their purchasing preferences. Targeted ads can also be placed in certain areas of the Web that are read only by specialist readers - analogous to the traditional placement of adverts in specialist magazines.

Traditional advertising media suffer from a long delay between the appearance of the advert and new purchases generated because of that advert. In the case of TV, this period may be more than one day. In the case of direct mail, this period may be over a week. Technology is emerging which will allow users to make an immediate purchase after seeing an Internet advert. This reduces this crucial waiting period and allows a direct relationship to be made between the advert itself and its success at marketing that particular product.

It is worth mentioning too that the Internet has already spawned new promotional models to keep readers visiting publishing sites: for example, interactive games.

Summary of the Invention

We have appreciated problems with computer networks as noted above. In particular, we have appreciated that existing networks and software cannot provide a specified service level without changes to software as noted above. Accordingly, in a broad aspect, the invention provides a method and apparatus for providing data communications at a specified level of service in a computer network.

In particular, the invention provides a method of communicating in a communications network having a first node, a second node and an

intermediate node, the method comprising: sending a request, from the first node, for data stored at the second node in the network; intercepting the request at the intermediate node; analysing the request to determine the data transfer parameters appropriate to the request; and supplying the requested data in the manner specified by those parameters.

The invention thus provides a specified level of service depending on an analysis of the request, in contrast to the best effort network which provides no specified level of service, or the known QoS protocols which require the program unit issuing the request to be aware of the level of service required. The invention advantageously provides a specified level of service in accordance with the data transfer parameters without any change to the application program unit or protocol at the node issuing the request.

In a preferred embodiment the network further comprises a descriptive data store for storing descriptive data, and the step of analysing comprises: retrieving the descriptive data from the descriptive data store; and analysing the descriptive data to determine the transfer parameters appropriate to the request.

The descriptive data store is a store to which reference is made to determine the transfer parameters appropriate to the request. The descriptive data store may contain data describing, amongst others, the nature of the data which may be requested, or the bandwidth allowed to any particular user or node. For example, the descriptive data comprises nature data describing the nature of the data requested, such that the step of analysing comprises: retrieving the nature data from the descriptive data store, and analysing the nature data to determine the transfer parameters appropriate to the request. The nature data may describe the file type of the data requested.

In the preferred embodiment the request is analysed to determine the nature of the data requested, and the data is supplied with the parameters appropriate to the nature of the data.

This feature advantageously provides an appropriate transfer of data according to the requirements of that data. For example, moving images are transferred with a high bandwidth, whilst text only requires a low bandwidth, high latency service. The data could be stored as a file, and the nature of the data requested determined as a result of analysing the file type. By "nature" of data, we mean the inherent structure of the data which depends on what the data represents e.g. text, sound, images or moving images. The file type could be denoted by the file extension.

In a preferred embodiment, the method further comprises determining the user initiating the request, and supplying the data requested in the manner appropriate to that user.

In the preferred embodiment, the invention further comprises analysing the request to determine which of two or more networks are appropriate to transfer the requested data, and transferring the data via the network selected as a result of the determination.

The embodiment of the present invention thus provides two further advantages over the problems outlined in the previous section as follows:

providing a device which is capable of transparently choosing between two or more computer networks to handle each request made by a client, in particular where said computer networks offer differing levels of service.

providing a device which delivers electronic content and targeted advertising to local users and can guarantee delivery of said content and advertising at a certain level of service to the person who is paying for delivery.

In the preferred embodiment, the response may also be analysed to determine the data transfer parameters appropriate to the request. Analysing the response can give further information to determine the file transfer parameters, such as the file type, file length, and even the preferred bandwidth of the transfer. This information in conjunction with the descriptive data may be used to determine the actual transfer parameters.

In addition to the methods noted above, the invention also provides a system for providing data communications at a specified level of service in communications network, the network having a first node and a second node, the network comprising: means for sending a request, from the first node, for data stored at a second node in the network; the system comprising: means for intercepting the request; means for analysing the request to determine the data transfer parameters appropriate to the request; and means for supplying the requested data in the manner specified by those parameters.

In the preferred embodiment, the second means comprises: a descriptive data store for storing descriptive data describing aspects of data stored at the second node in the system, means for retrieving the descriptive data describing the requested data from the descriptive data store; and means for analysing the descriptive data to determine the transfer parameters appropriate to the request.

The descriptive data store holds any data which may be used to determine the appropriate transfer parameters, such as the nature of the data, the users' permitted bandwidth, and so on.

The invention also provides methods and apparatus for analysing the response to a request in a similar manner to determine the data transfer parameters, in conjunction with the analysis of the request, or separately. For example, the response may be analysed to determine two 'key elements', such as the MIME-Type and the Content-Length fields of

an HTTP header. The former indicates the type of data included in the response, while the latter indicates the length of the file being delivered.

Description Of Embodiments Of The Invention

Embodiments of the invention will now be described, by way of example only, and with respect to the figures, in which:

Figure 1 is a schematic view of a computer network;
Figure 2 is a schematic view of a network using a proxy cache;
Figure 3 is a graphical representation of current media alternatives;
Figure 4 is a schematic view of a first embodiment of the invention;
Figure 5 is a flow diagram showing steps of the first embodiment;
Figure 6 is a schematic view of the interrelationship of the various services provided by the first embodiment;
Figure 7 is a block diagram of the call admission stage of the first embodiment;
Figure 8 is a block diagram of the select QoS stage of the first embodiment;
Figure 9 is a block diagram of the achieve QoS stage of the first embodiment;
Figure 10 is a block diagram of the software components comprising the first embodiment of the invention;
Figure 11 is a schematic view of a computer network according to the second embodiment of the invention;
Figure 12 is a flow diagram showing steps of the second embodiment;
Figure 13 is a schematic view of the interrelationship of the various services provided by the second embodiment;
Figure 14 is a block diagram of the select QoS stage of the second embodiment;
Figure 15 is a block diagram of the achieve QoS stage of the second embodiment; and

Figure 16 is a block diagram of the software components comprising the second embodiment of the invention.

The invention will be described with respect to two embodiments. In the first embodiment, the intermediate node is located between the client nodes from which requests are issued and the wide area network (or networks) over which the request is carried to the node from which the data is requested. In this embodiment the device embodying the invention is called the Integrated Switch Gateway (ISG). In the second embodiment, the intermediate node is located between the wide area network and the node from which the data is requested, and can be additionally located between the node issuing the request and the wide area network. In the second embodiment, the device is known as an Application Protocol Classifier (APC).

In both embodiments a node (ISG or APC) is provided at some point in a communications network which intercepts data requests, analyses the requests and provides data delivery with specified parameters such as bandwidth, latency, jitter and burstiness, thus providing a specified service level. In either embodiment, the user is provided with a guaranteed quality of service.

First Embodiment

In the first embodiment, the Integrated Switch Gateway (ISG) is a computer running various system program units which together enable it to act as a proxy or proxy cache for a number of IP-based services. A typical ISG computer will be a server-class computer, e.g. having one or more fast central processing units, a large main memory store and a large reliable on-line disk store. The first implementation of the ISG runs on Intel Pentium computers under the UNIX operating system using ATM

(Asynchronous Transfer Mode) networks and is written in the C and Java languages.

The function of the ISG is to intercept requests from a first node for analysis to determine the QoS parameters required for the data communication. In this embodiment, the supported QoS parameters are simply bandwidth and latency, though the design can be extended without difficulty to cover other parameters such as jitter.

The ISG includes a store holding descriptive data, a portion of which is known as the metadata cache, which stores information in the form of metadata relating to data stored at other nodes on the network. A portion of data in the form of video, sound, text or of other nature is known as an "object". Metadata is a short description of an object or group of objects at a particular content server node, and includes a description of the QoS parameters for the object(s). The QoS appropriate to any particular request is determined with reference to the metadata for the requested object. Metadata is distributed throughout the network by the policy service which will be described later.

The ISG is connected to a group of clients via direct or indirect network links. Each network link between a client and the ISG may be a direct link or may go through one or more other nodes. Each of said links is referred to as the client access network. The access network may be a telephone subscriber line, cable television network, local area network, home or small office network or some other private network owned or leased by the client or client's organisation.

A typical client machine will be a personal computer or network computer equipped with an application such as a World Wide Web (WWW) browser and an IP stack with a connection to the ISG which may be by analogue modem, cable modem, ISDN, xDSL, LAN, wireless modem or other common connection method. This connection to the ISG is termed the client access network, and it is assumed that this network is either dedicated to a single

client machine or that it can provide a fixed minimum bandwidth and maximum latency to this client machine.

The ISG is further connected to one or more wide area networks. Each of these wide area networks may be a best effort network or a QoS network or a network which offers both best effort and QoS services. A best effort service could, for example, be provided over a QoS network.

If the ISG is connected to more than one wide area network, then the ISG can perform both the transparent choice between networks and the delivery of electronic content and advertising to the clients. If the ISG is connected only to a best effort network, then clearly the ISG cannot function as a means of transparently choosing between alternative networks; it can, however, still deliver electronic content and advertising from local storage to the clients with as high a level of service as the client access network can deliver (often significantly higher than the per-user wide area network connection bandwidth). In this embodiment, the content itself is replicated from servers to the ISGs over the best effort network periodically in batch mode.

An alternative embodiment could provide data from the cache with a specified QoS, by simply modifying the ISG decision flow outlined in fig. 12 to perform the check cache stage 120 immediately after the select QoS stage 140, rather than before.

The metadata distribution service is responsible for both collecting metadata from content servers and delivering said metadata to ISGs. The providers of the content servers may annotate services with explicit metadata information, or they may run programs periodically on the server which automatically analyse content and provide the metadata information, or they may do nothing in which case the metadata distribution service will periodically scan the contents of the server to determine metadata. The metadata distribution service will make metadata information for all sites available and the ISGs will download this metadata periodically.

The ISGs will normally receive only the difference between the current metadata information and the previous metadata information downloaded in order to reduce the time and cost of downloading this information.

Fig. 4 shows a specific example of the system described in the previous paragraph. In this specific example, there are four client nodes 10. These client nodes have direct connections to the ISG computer 60. The ISG in this example is connected to two physical wide area networks. The first physical network offers QoS guarantees 70 and the second physical network offers best effort service 80. Also shown in this figure is a server node 50 which is connected in this example to both wide area networks. Notice that fig. 4 is an example only and is not meant to imply that the ISG will only be used in this configuration. In particular, it is not required that server nodes are connected to both a QoS and a best effort network. It is possible that there will be servers that are connected only to a QoS network or only to a best effort network.

Typically, the ISG and associated nework services are provided by an organisation called the "ISG service provider". This organisation is responsible for administering the ISG, associated network services,and (optionally) the wide area neworks. Frequently this organisation will be an Internet service provider or telecommunications company.

A specific embodiment of the software components that make up the ISG in our design is shown in fig. 10. In fig. 10, the rectangular boxes refer to logical software components which constitute our design and the circles refer to major data stores in our design. Components with dotted (rather than solid) outlines refer to further features which could be added to an embodiment of the invention, but which will not be described in detail herein.

The ISG is transparent to the client software, thus ensuring that the client software and client hardware do not need to be upgraded or changed

in any way. To ensure this transparency feature, one of the following two requirements is preferably met:

The ISG is located at a point in the network where all paths from each client to any server that requires QoS guarantees pass through the ISG. Fig. 4 shows a specific example where this method has been used.

The ISG modifies the routing process of the network to ensure that all requests from each client to any server that require QoS guarantees pass through the ISG. IP offers several features which allow this to be done without modification of the client and thus this method can be used on IP-based networks; similar approaches are possible on many other types of network. As a concrete example, it is possible on many IP-based networks for the ISG to advertise routes to remote sites that are available on a QoS network using a standard routing protocol, e.g. the Routing Information Protocol (RIP). Internal IP routers in the client access network will then transmit data packets intended for said remote sites to the ISG, and will transmit other packets directly to the best effort Internet. Any packets which were unintentionally diverted to the ISG would be forwarded by the ISG to the best effort network after making the QoS decisions outlined below.

The ISG may receive requests from any client. Fig. 5 is a flow diagram showing the steps taken by the ISG upon receipt of a request. After accepting a request the ISG goes through up to four main stages before either replying to the request, rejecting the request or initiating a network connection to the server to satisfy the request. The four stages are: call admission 110, check cache 120, select QoS 140, achieve QoS 160, as follows:

Call Admission

The call admission stage 110 from fig. 5 is expanded in detail as a data-flow diagram in fig. 7. The purpose of the call admission stage is to determine whether or not the client machine which has submitted the request has permission to submit said request. The said permission may comprise one or more of the following criteria, normally applied in the order used here. All data mentioned in these criteria is configured by (1) the administrators of the content servers, for content site data (2) the administrators of the ISGs, for ISG subscription data. The data is input using graphical administration tools, then replicated to all ISGs by the metadata distribution service.

The criteria used in this process are as follows:

A 'content server confidentiality agreement list' is retrieved 111 from the database of QoS aware sites 114 - this list is composed of tuples of the form (ISG name, content server name), detailing all ISGs permitted to access information from specific content servers. If the ISG name determined by the second authentication stage (below), when combined with the content server name extracted from the request, does not match a tuple in this content server confidentiality agreement list 111, the ISG rejects this request quickly, without needing to check the user identifier. If a match occurs, the ISG proceeds to the next stage of call admission.

A 'content subscription-based level of service agreement list' is retrieved 112 from the database of subscribed users 115 - this list consists of tuples of the form (user identifier, content server name, content subscription QoS), detailing which users are permitted to access which content sites and at what QoS, based on the content site subscription they have purchased. If the user identifier determined by the first authentication stage (below), combined with the content server to which the request is addressed, does not match the corresponding values of a tuple in this list, the ISG rejects

this request. If a match occurs, the ISG proceeds to the next stage of call admission.

any other criteria as decided by the ISG owner and which may be implemented as a part of call admission control 113.

In order to perform call admission in the context of the "real world" untrusted network(s) to which the ISG will be connected, two authentication stages are required.

The first of said authentication stages is used to securely associate a particular human user with the machine client that makes the request on the said user's behalf. To perform said authentication the user can be required to type in a user identifier and a secret password at the start of each session of request-reply transactions. This stage retrieves an 'ISG subscription list' from the database of subscribed users 115 - this describes users' subscriptions to the ISG as opposed to content sites, and is distinct from the 'content subscription-based level of service agreement list'. The 'ISG subscription list' takes the form of a list of tuples of the form (user identifier, password, subscription end date, ISG subscription QoS, measured access network QoS). If a tuple matching the supplied user identifier and password is found, the user is considered to be authenticated; if not, the user has failed the authentication process and the overall call admission process fails.

For client access networks that have a variable QoS (e.g. cable modems, analogue modems, etc), the ISG will normally measure the actual access network QoS at the first time that a user undergoes the call admission process, and periodically thereafter, and store this information in the 'measured access network QoS' field of the database of subscribed users 115. The client access network QoS is measured with minimal overhead by recording the actual latency for a small number of measurement packets of two differing sizes, S1 and S2, sent from the ISG to the client machine, where S1 is the smallest length possible on that network and S2

is the greatest packet length allowed on that network. The ISG then computes the following:

L1 = average latency for packets of size S1
L2 = average latency for packets of size S2
B = (S2 - S1) / (L2 - L1)

The calculated bandwidth B is a useful approximation of the actual bandwidth available on the client access network, while L2 is a useful approximation of the maximum latency, assuming no queuing effects due to congestion. Since the ISG carefully manages the bandwidth of responses actually using the client access network for each client, and the client access network has sufficient capacity to support ISG subscriptions available to that point, the latter assumption is reasonable. Thus, the ISG uses (B, L2) as the 'measured access network QoS' field for the tuple describing this user in the database of subscribed users 115.

For client access networks that have fixed QoS (e.g. ISDN), the measured access network QoS field is simply set by the administrator to a standard value based on the capabilities of the client access network.

At the time of selling a user the ISG subscription, the 'permitted QoS' field of the 'ISG subscription list' in the database of subscribed users 115 is set to a suitable value.

The second of said authentication stages is used to securely identify the ISG system to the content sites. The purpose of this second authentication stage is to prevent unauthorised use of the content site by client machines masquerading as ISGs. To perform this second stage of authentication the ISG and content server can go through a standard challenge-response authentication process whenever a new connection is set up from the ISG to the server. This stage uses the database about QoS aware sites 114.

Check Cache

The ISG includes a data cache for storing frequently accessed data to facilitate speed and QoS of data retrieval. Thus, for requests in any protocol where the responses are cacheable (e.g. HTTP in IP networks), the ISG checks the data cache before sending a request across a network for data in the form of an object. When performing the cache checking stage 120 the system uses the name of the object requested as an index into a 'cache table' of (object name, object response) tuples. If the said named object exists in the table then the request may be finished by sending the associated object response back to the client as shown at 130 in fig. 5. This 'cache table' is built up by storing responses to earlier requests as they pass through the ISG.

The object response field may not contain the complete response information in the case where another client has earlier requested the same object and is currently receiving the object response. In this case, the object response field will be dynamically completed by the other client while the current client is loading the first part of the response. An example of the use of this capability would be to ensure that if one client requests a ten minute long stored video file, and ten seconds later a second client requests the same video file, the ISG can provide caching to the second client, facilitating delivery of the video file at the required QoS without requiring a second network connection.

If the said named object does not exist in the table, then the ISG continues to the select QoS stage 140.

Regardless of whether the object is delivered from cache or through a network from another node, the ISG ensures that the transaction is logged, as described below.

Select QoS

The select QoS stage from fig. 5 is expanded in fig. 8 as a data flow diagram. As throughout the ISG, QoS is defined as a tuple of the form (bandwidth, latency), with two possible special values of bandwidth: -1 indicates that best-effort service should be used, and 0 indicates that the traffic should be blocked. When performing said QoS selection stage, the ISG will use the following three available sets of data in order to come to a decision:

information about the bandwidth and latency of the client access network including, if necessary, information about the remaining capacity of the client access network after taking into account other responses that may be currently proceeding over that access network 141,146. The ISG uses the 'measured access network QoS' field (taken from the 'ISG subscription list' tuple for this user in the database of subscribed users 145), to establish the total bandwidth available, then subtracts, for each response currently proceeding through the ISG to this client node, the bandwidth element of the QoS selected for that response. The result is called 'available access network bandwidth' and is held by the ISG for later use in this QoS selection stage. We assume latency is fixed for the client access network, since the ISG is carefully managing the load on this network, avoid most queuing effects; however, guaranteeing client access network QoS is outside the scope of the ISG.

information about the user's subscription level and subscription service time remaining if applicable 142, 145. In order to determine the user's subscription level of service, ISG first retrieves the 'ISG subscription QoS' from the tuple corresponding to this user in the database of subscribed users 145. Next, the ISG retrieves the 'content subscription-based level of service agreement list' tuple corresponding to this user, and extracts the 'content

subscription QoS' field. The ISG combines these two fields by taking the minimum bandwidth of the 'ISG subscription QoS' and the 'content subscription QoS', and the maximum latency of these two QoS values, resulting in a single 'permitted QoS' value of the form (bandwidth, latency).

Now that the subscription-derived QoS has been established, the ISG modifies this to take account of the client access network over which the user is currently connected (which allows simple handling of examples such as mobile users connecting via modem while travelling and via ISDN while at home). To accomplish this, the ISG modifies the bandwidth element of this permitted QoS, setting it to the minimum of its current value and the 'available access network bandwidth' calculated earlier (as explained in the Call Admission stage). The ISG also modifies the latency element of this permitted QoS, setting it to the maximum of its current value and the latency element of the 'measured access network QoS' field of the database of subscribed users 115. The result of these two modifications is termed the 'allowed access network QoS', since it represents not just what is permitted by subscription but is possible given the limitations of, and current traffic using, the client access network. This 'allowed access network QoS' represents the maximum bandwidth and minimum latency that can be supported by the user's subscription when combined with the client access network currently in use.

In order to determine the user's ISG subscription service time remaining, the ISG retrieves the 'subscription end date' field from the 'ISG subscription list' tuple corresponding to this user in the database of subscribed users 145. If this date is less than the current date, indicating that the subscription service time has expired, the ISG sets the maximum bandwidth to zero, effectively refusing QoS service to this traffic.

metadata information about the requested object 143, describing the QoS required to deliver this object.

The metadata cache 144 holds metadata information locally on the ISG about a large proportion of all remote objects. Metadata is a short description attached to an object or a group of objects at a particular content server and specifying at least the QoS parameters for the object(s). The size of the total metadata store is kept small by using two strategies:

Most metadata information is kept in the form of rules, which are intended to summarise the metadata requirements for a whole group of objects in a single area of a site, on a site as a whole or for any site connected to a particular QoS network. As an example, for HTTP requests over IP, a single rule "*.html" applying to all sites connected to a QoS network would give a particular low guaranteed bandwidth for all HTML files on those sites. A particular site could then specify a general rule "*.mov" giving a default bandwidth for all movie files on the site, but override this general rule with extra metadata information for particular movie files that require a different level of service from the default. Another HTTP example illustrates how it is possible to describe all objects in a single part of a site - a single rule such as "purchase/*" could give high QoS to all objects under a single directory, in this case those concerned with allowing customers to place orders via the network.

The actual metadata information itself is typically rather small - of the order of 100 bytes per object or per rule. Therefore a million or more such metadata entries may be stored on each ISG without requiring particularly large or costly stores.

The metadata information about objects at each content server is periodically push-replicated by the metadata distribution service to all the ISGs, the aim of said process being to ensure that the ISGs have at each point in time metadata about a very large proportion of all objects available on the ISG locally. Metadata about objects is stored in the ISG's metadata cache 144.

In the rare cases where metadata about a particular request object is not available in the ISG's own metadata cache, a fall-back method is employed to attempt to estimate the QoS parameters required by the requested object. One or more of the following fall-back methods may be employed to derive an estimate of the required QoS - the choice of which method(s) will be used is entirely up to the administrator of the ISG service, based on the content sites accessible and the completeness of the metadata provided by them:

derive the QoS directly from the characteristics of the request, using default metadata rules that are built-in to the ISG. For example, for IP requests, the destination port number can be used to make a simple determination of QoS required, using a default rule mapping this port number to a QoS. Another example would be that HTTP URL (Uniform Resource Locator) could be used to extract a file extension field such as ".txt" which can be used to determine a suitable QoS requirement for the response, using a default rule based on file extension.

use the average QoS requirements for previous objects of this same type requested from this particular server, where in this context, type refers to either the HTTP URL file extension field, or to the destination port number, or to the Internet MIME type if that can be determined.

use the default forwarder service, described later, to issue a short request (e.g. "GET-HEAD" in the HTTP protocol) in order to retrieve

the first part of the response and determine from that first part the QoS required. For example, with the HTTP protocol, the MIME-Type and Content-Length fields could be used as parameters for a default rule mapping these two fields to a QoS.

assume the response requires a constant bandwidth connection at the rate of the client access network.

assume the response requires a constant bandwidth connection at the available bandwidth of the client access network (as calculated above), or at some proportion of this bandwidth.

assume the response requires only "best-effort" service.

The 'object QoS', whether determined from the metadata or from the fall-back rules, is combined with the 'allowed access network QoS' derived above, in order to provide a 'desired QoS' value. This 'desired QoS' refers only to the QoS that must be provided by the wide area network, not to the QoS provided by other elements such as the client access network or the content server itself. The steps in calculating the 'desired QoS' are:

First, the bandwidth element of the 'desired QoS' is calculated as the minimum of the bandwidth element of the allowed access network QoS and the bandwidth element of the object QoS. This ensures that neither the bandwidth available in the client access network nor that allowed by the user subscription is exceeded.

Second, the latency element of the 'desired QoS' is calculated by subtracting the latency element of the allowed access network QoS from the latency element of the object QoS. This ensures that both the latency possible with the client access network and the latency allowed by the user subscription are accounted for, by reducing the acceptable wide area network latency by a corresponding amount.

The result of this 'desired QoS' calculation may indicate that the object QoS cannot be provided, owing to the user subscription, client access network or current traffic on the client access network. In particular, either a desired QoS bandwidth that is less than the object QoS bandwidth, or a desired QoS latency that is less than or equal to zero, indicate that the desired QoS is impossible, in which case the ISG sets the bandwidth element of the QoS to -1, indicating best-effort. The request is then handled via the 'achieve QoS', where the '-1' bandwidth will cause the transaction to use the best-effort network.

Achieve QoS

The achieve QoS stage in fig. 5 is expanded in fig. 9 as a decision tree. The aim of said stage is by incorporating the information collected in stages 110 and 140 to come to a sound decision about the most efficient way to achieve the quality of service level settled upon in these preceding stages (i.e. the 'desired QoS' value).

The achieve QoS stage proceeds as follows:

1. Decide whether the desired QoS level may be achieved using a best effort 161 or QoS 162 network. The ISG inspects the bandwidth element of the 'desired QoS' - if this is -1, indicating best-effort service, the best-effort network is used and the remaining steps are skipped.

2. If the bandwidth element of the desired QoS is greater than zero, the ISG decides whether the desired QoS level can be satisfied in one of the following ways:

   a) entirely through the default forwarder service 163 - this is used where the expected response is short - specifically, where

the total latency of the entire expected response is less than or equal to 150% of the time to set up a new connection through the QoS network. This ensures that short request/response transactions are processed quickly without undue network overhead.

b) by routing the first section of the response through the default forwarder service 163 and at the same time establishing a connection through a QoS network for the remainder of the response 164 - this is used where the response is lengthy (i.e. greater than 150% of the time to set up a new connection through the QoS network) and the latency part of the desired QoS level is small. A small latency is defined as less than or equal to 150% of the time to set up a new connection through the QoS network.

c) by establishing a connection through a QoS network for the whole response 164 this is used where the response is lengthy (i.e. greater than 150% of the time to set up a new connection through the QoS network) and the latency part of the desired QoS level is large. A large latency is defined as greater than 150% of the time to set up a new connection through the QoS network.

If during the achieve QoS stage it is determined that the level of QoS required cannot be achieved, then the request should be cleanly rejected. This determination that the QoS required cannot be achieved may come from the QoS network on attempting to set up a connection (in the form of a network error indicating that the connection could not be set up); it may also come from the default forwarder service, if it determines that the ISG is overloading the default forwarder.

The rejection resulting from this type of network error can take the following form. The ISG sends an 'alert page' to the user explaining that the desired QoS cannot be achieved and, if possible, giving a brief

explanation of why this is so. This 'alert page' may also offer a button which, if pressed, will allow the user to download the requested information using a non-guaranteed or best effort network. The user should be billed for the service they actually use, so in this situation, the user would be billed for using only the best effort service. In some cases it may not be possible to offer said best effort option - for example when a particular piece of information is only available by a QoS network.

Where the user request is in HTTP, the alert page will simply be an HTML (Hypertext Markup Language) page, replacing the expected response in the user's web browser. Where the request uses some other protocol, the ISG may either reject the request without explanation, or may optionally send an HTML page to the 'alerter', a small application program unit residing on the client machine. This 'alerter' then displays the HTML page and allows the user to interact with it as normal.

Initiating the wide area network transaction with the desired QoS

After the achieve QoS stage 160 from fig. 5 a wide area network transaction may be initiated given the condition that the object required cannot be found in the ISG's local cache. The wide area network transaction will take place as decided by the algorithms employed in the achieve QoS stage 160. The response may be cached as it is received. Large files and open-ended multimedia streams are not generally cached. Furthermore, it is possible for a server to defeat caching of a particular object as part of the HTTP protocol.

The ISG is aided by two well-known network services, and one ISG-associated network service, as shown in fig. 6.

One of said network services used by the ISG is the billing and usage logging service 185, which is used by the ISG service provider to generate bills for customers. The billing and usage logging service may

be available to the ISG over exactly one of the best effort or QoS networks which are connected to the ISG. The billing and usage logging service is explained in greater detail in the section "billing and usage logging service" below. Although it would be possible to conceive of an ISG which is not connected to at least one network that has a billing and usage logging service, such an ISG would not be able to perform billing, usage logging and a number of other desirable services which are preferable for the current embodiment of the invention.

The second of said network services required by the ISG is the default forwarder service 187. The default forwarder service (often also known simply as the default forwarder) must be available where one or more QoS networks is connected to the ISG. In the case where the ISG is not connected to any QoS network, a default forwarder is not required. The default forwarder is explained below in more detail in the section "default forwarder service".

One network service which is not well-known and is closely linked to the ISG is the metadata distribution service 186, previously described. The metadata distribution service may be available to the ISG over exactly one of the best effort or QoS networks which are connected to the ISG.

It is possible to conceive of a single logical or physical computer system which performs more than one of the three preceding services. It is also conceivable that a single logical network service will be performed by more than one physical computer system, which thing might be particularly beneficial to reduce the average distance covered by transactions between ISGs and network services or to allow the service to adequately cope with the volume of transactions made by a large population of ISGs.

Network usage and transaction logging

It is important that the ISG logs transactions which use precious, scarce or costly resources such as the QoS network or which access information that has been made available by the content publishers at a cost or which relate to advertising material or which are to be made available for usage tracking. To ensure this, the ISG logs all transactions where (a) their metadata specified a QoS with bandwidth greater than zero or (b) they were served from the ISG cache. It is the responsibility of the content site to ensure that there is correct metadata for any object for which transactions are to be logged, and that this metadata specifies a QoS with a bandwidth element greater than zero. This scheme ensures that all QoS traffic is logged, even if cached, without the excessive overhead of logging best-effort traffic. These transaction logs are periodically submitted automatically by the ISG to a central billing and usage logging network service.

By logging transactions in the manner outlined above, the embodiment of present invention will allow flexible and precise billing to be done, based upon both the cost associated with the network connection itself and the cost of the information being delivered.

The said central server will be able to collect content usage counts for content sites and advertising material and will be able to deliver these in an audited form to the relevant parties. Whereas current content sites attempt to defeat caching in order to collect content usage count information, with the present invention there will be no reason to defeat caching. Moreover, these independently audited content usage counts will carry more weight with advertisers. The effect of the present invention will be three-fold: to reduce wide area network load, to reduce content site load and to provide audited content usage count figures.

Because the system outlined in the present invention will maintain a subscriber relationship with clients, it will be able not only to supply raw content usage counts as in the preceding paragraph, but also usage analysis based on more complex demographic information. Some specific

examples of the type of more complex usage analysis that might be delivered are: a histogram of the age of readers of a particular piece of published content; a chart of the demographic background of readers against the content on a single server; or a histogram of the incomes of subscribers of all financial services offered on the QoS network.

Billing and usage logging service

The billing and usage logging service is a network service that is responsible for translating the transaction logs submitted periodically by the ISGs into a form required by the infrastructure or the agent that bills for the premium service.

Default forwarder service

The default forwarder is a network service that is implemented using guaranteed quality network connections to each ISG and server that is available on the QoS network. It may be implemented as part of the QoS network (using guaranteed QoS connections between all ISGs and servers) or as a network of standard IP routers. The default forwarder service just forwards arbitrary data from any connected node to any other connected node. Any connected node that wishes to use the default forwarder service to communicate with any other node on the same QoS network simply sends a message containing the address of the destination node and the data to be sent.

The default forwarder service avoids the latency of setting up calls over QoS networks: in a typical QoS network, setting up a connection requires minimally that all nodes and links along the path of the proposed connection be examined and so this call set-up time can be quite large - of the order of 0.5 seconds for a large wide area network. To avoid this large latency, the ISG can if necessary route the first part of the request and response data over the default forwarder whilst simultaneously setting up the connection to carry the major part of the

data. Once the connection has been established, the data transfer is instantaneously switched from the default forwarder path to the connection path.

The default forwarder service is intended to be used to avoid the latency of setting up connections over QoS networks, and not for transmitting large quantities of data. Thus the demands made of this service should be modest. Nevertheless, it is possible that the default forwarder service would need to be replicated or hierarchical in very large QoS networks.

Second Embodiment

In the second embodiment, the Application Protocol Classifier (APC) is a computer running various system program units which together enable it to act as a proxy or proxy cache for a number of IP-based services. One possible APC computer is a server-class computer, e.g. having one or more fast central processing units, a large main memory store and a large reliable on-line disk store: this configuration will serve a larger user base and provide proxy caching. Another possible APC computer is based on a network device (such as a router or firewall) rather than a general purpose computer, and has only a small on-line disk store, or no disk store: this will serve a smaller user base and will not itself provide proxy caching. It is also possible for the APC to be located wholly on a host computer (a client or server machine): this is advantageous where the bandwidth used per host computer is high, or where only certain machines require APC services. The APC thus corresponds closely to the ISG in the first embodiment of the invention. The first implementation of the APC runs on Intel Pentium computers under the UNIX operating system, using IP networks employing RSVP (Resource Reservation Protocol) and is written in the C, C++ and Java languages.

The function of the APC is to intercept requests and responses from a first node for analysis to determine the QoS parameters required for the data communication. The ISG in the first embodiment only analyses requests, and is always located between the client and the wide area network. By contrast, the APC analyses both requests and responses, and two APCs are involved in each transaction, located (1) between the client and the wide area network and (2) between the server and the wide area network.

In general, machines served by the APC may be clients or servers, or both simultaneously - the term "host" is used as a general term for a machine that may be a client or a server. Where the

terms "client" and "server" are used in this section, they denote only the role taken by a host in a particular transaction.

One key difference from the ISG embodiment is that, where the server machine may require QoS to deliver its response to a client, the 'server-side' APC (i.e. the APC closest to the server) is responsible for selecting the appropriate QoS and achieving this QoS, possibly by setting up a wide area network connection to the client-side APC. In the ISG embodiment, there is no server-side ISG and any wide area network connections are therefore set up from the client side; the ISG embodiment also requires the server to be directly connected to the wide area network. The APC design removes the need to communicate information about QoS requirements from all servers to all client-side APCs.

The APC embodiment also fully supports symmetric use of QoS, i.e. the situation where two communicating hosts both need to send data with guaranteed QoS. The ISG embodiment could be extended to support symmetric QoS but does not support this as standard. An example of the use of symmetric QoS is a two-way video conference, in which both hosts require QoS in order to transmit video at high quality.

In this APC embodiment, the supported QoS parameters are simply bandwidth and latency, though the design can be extended without difficulty to cover other parameters such as burstiness and jitter.

The APC includes a policy cache (known in the ISG embodiment as a metadata cache) which stores descriptive data such as information in the form of policy data relating to data stored at other nodes on the network. This policy data is obtained from the policy service, described below. A portion of data in the form of video, sound, text or of other nature is known as an "object". Policy data is a short description of an object or group of objects at a particular content server node, and

includes a description of the QoS parameters for the object(s). The QoS appropriate to any particular request is determined with reference to the policy data for the requested object. Policy data is distributed throughout the network by the policy service which will be described later.

Each APC is connected to a group of hosts (i.e. client and server machines) via a "host access network". This network corresponds to the client access network in the ISG embodiment, but has been generalised to cover both client and server machines. Apart from this, the host access network is identical to the client access network.

As with the ISG embodiment, each APC is further connected to one or more wide area networks. Each of these wide area networks may be a best effort network, a QoS network, or a network which offers both best effort and QoS services.

If the APC is connected to more than one wide area network, then the APC can perform both the transparent choice between networks and the delivery of electronic content and advertising to the clients. If the APC is connected only to a best effort network, then clearly the APC cannot function as a means of transparently choosing between alternative networks; it can, however, still deliver electronic content and advertising from local storage to the clients with specified levels of service. In this embodiment, the content itself is replicated from servers to the APCs over the best effort network periodically in batch mode.

The policy service is similar to the metadata distribution service in the ISG embodiment, but with improvements to its scalability and manageability. This service is responsible for delivering policy data to APCs; this policy data corresponds to the ISG embodiment's metadata, network data and user data. The providers of the content servers use graphical administration tools to enter and maintain policy data; they run programs periodically on the server which automatically analyse content

and provide the policy data, but these programs are outside the scope of the APC. The policy service makes policy data for all sites available and the APCs download this policy data on demand, i.e. whenever this data is required to make a decision on requests or responses being processed by the APC. This design is significantly more scalable than that used by the ISG embodiment because (1) most policy data remains close to the server supplying the actual content data, rather than requiring every APC to have complete policy data for all servers and (2) data is downloaded to the APC only on demand, so the size of the policy cache on the APC is related to the activity of its users, rather than to the complexity of the entire network. The improved manageability of this design is due to the ability to centrally coordinate all policy data rather than defining it largely on a per-server basis.

Fig. 11 shows a specific example of the system described in the previous paragraph. In this specific example, there are four client nodes 210 and two server nodes 220. The client nodes have connections to the first APC 230 via a host access network 240 using, in this example, dedicated links, while the server nodes have connections to the second APC 250 via a second host access network 260, which in this example does not use dedicated links. Both APCs in this example are connected to two physical wide area networks. The first physical network offers QoS guarantees 270 and the second physical network offers best effort service 280. Notice that fig. 11 is an example only and is not meant to imply that the APC will only be used in this configuration. In particular, it is not required that an APC is connected to both a QoS and a best effort network. It is possible that there will be APCs that are connected only to a QoS network or only to a best effort network. Also, it is possible to connect both client and server nodes to a single host access network, and for each APC to serve a mixed population of client and server nodes.

A specific embodiment of the software components that make up the APC in our design is shown in fig. 17. In fig. 17, the rectangular boxes refer to logical software components which constitute our design and the circles refer to major data stores in our design. Components with dotted (rather than solid) outlines refer to further features which could be added to an embodiment of the invention, but which will not be described in detail herein.

The APC runs on a computer system (termed the 'APC platform') with some important attributes. First, this system includes full support for TCP/IP communications, known as a 'TCP/IP stack'. Secondly, this system supports 'traffic shaping', i.e. the TCP/IP stack is capable of sending out IP packets at different rates, depending on how they are classified into two or more queues. Each queue within the traffic shaping software has a specified rate (i.e. the bandwidth it is permitted to use). Through careful control of traffic shaping queues and their parameters, it is possible for the APC to ensure that the underlying TCP/IP stack does not cause the QoS of application requests/responses to vary from that selected by the APC. Although an APC which is very lightly loaded may be able to preserve QoS characteristics of traffic without using traffic shaping, it is important to use traffic shaping under higher loads of QoS traffic. Traffic shaping is well known to those skilled in the art, and need not be described in detail here. The embodiment of the invention may use any known system for providing control of the transfer parameters once the appropriate parameters have been selected. One implementation of traffic shaping is Class Base Queueing (CBQ), a publicly available implementation described in Floyd, F., and Jacobson, V., Link-Sharing and resource management models for packet networks, IEEE/ACM transactions on networking, vol 3, no 4, p's 365-386 August 1995. A general introduction to the whole area of traffic shaping can be found inn An engineering approach to computer networking, Addison-Wesley professional computing series by S. Keshev, both references are incorporated herein by reference.

The APC is transparent to the host software (both application program units and system program units), thus ensuring that the host software and host hardware do not need to be upgraded or changed in any way. To ensure this transparency feature, one of the following two requirements is preferably met:

The APC is located at a point in the network where all paths from any host to any other host that may require QoS guarantees pass through the APC. Fig. 11 shows a specific example where this method has been used.

The APC modifies the routing process of the network to ensure that all requests from any host via the wide area network(s) that require QoS guarantees pass through the APC. This is done as described for the ISG embodiment.

The APC may receive requests and responses from any host. While the ISG embodiment only analyses requests, the APC can analyse requests or responses, depending on the characteristics of the protocol. This provides more flexibility (since some useful information for choosing QoS is present in responses).

Fig. 12 is a flow diagram showing the steps taken by the APC upon receipt of a request or response. After accepting a request or response 310, the APC goes through up to four main stages before either (for requests) responding to the request from a data cache, rejecting the request/response or forwarding the request/response to its destination. The four main stages are: call admission 320, select QoS 340, check cache 360, achieve QoS 380. Following the achieve QoS stage, the APC processes the request/response by forwarding it to its destination, and logs the transaction.

The four main stages are described below:

Call Admission

This section describes the call admission stage 320 from fig. 12. Depending on the network configuration, the call admission stage in this embodiment (1) identifies a host that is authorised to request QoS services and/or (2) identifies a human user that is similarly authorised. Later stages deal with determining exactly what QoS this host or user is authorised to request.

In the ISG embodiment, the ISG also checks that the user and client machine have permission to use QoS services with a particular remote server. This step is required in the ISG embodiment because network connections are set up from the client side of the wide area network, but charged to the content server; care must therefore be taken to avoid incurring unauthorised costs to the server owner by setting up a connection for an unauthorised client. In the APC embodiment, however, the server-side APC sets up any wide area network connections that may be necessary; these connections will normally be charged to the server owner, who will be a customer of the APC service provider, preventing this type of fraud by unauthorised clients.

Where the APC is used to provide QoS for delivery of data from servers to clients(i.e. the asymmetric QoS case), it is only necessary to authenticate the server, since requests do not require high QoS. The current embodiment therefore focuses on server authentication - since servers typically do not use dial-up links to connect to the APC, a simple mechanism based on network address is adequate. The APC does provide an extension facility for support of more complex authentication mechanisms, which are important in the symmetric QoS case where both clients and servers require QoS, e.g. for videoconferencing.

Whatever the type of authentication, users and hosts which pass authentication are ultimately mapped into a unique 'policy account' corresponding to the user or host.

In the common case of a host that is connected via a fixed host access network, rather than dial-up (e.g. a client or server in

a corporation's network connected via a leased line to an ISP), the APC simply performs call admission on the host as follows:

The APC accepts the request or response, and extracts the host's IP address, which uniquely identifies it and is generically termed an 'authenticator'.

The APC consults its local policy module (LPM) in order to map the IP address into a 'policy account' (using a table mapping IP addresses to policy account IDs). If no such policy account exists, the request/response is rejected, otherwise call admission succeeds. The Policy Account tuple, which includes the account name and account ID, is kept by the APC for future use in select QoS 340.

Using the IP address as an authenticator is acceptable where the APC and the hosts connected to the host access network are considered secure against 'IP spoofing' attacks by intruders, in which an unauthorised host sends packets that are ostensibly from an authorised host. Where this is unacceptable, IP-level security such as the use of the IPSEC (IP Security) Authentication Header standard can be implemented between the host and the APC, and this authentication can be used instead of the IP address.

In the case of a host that is using a dialup host access network (e.g. a client using an ISDN link into the service provider's network), the APC performs call admission on the user of the host, rather than simply on the host, since the service is potentially accessible to intruders via the dialup connection. This type of authentication is an example of extending the APC to more complex authentication mechanisms. This embodiment can easily be interfaced to RADIUS, a common authentication protocol for dialup access to IP and other networks, but is not in any way limited to this authentication mechanism.

When RADIUS is to be used for authentication, a third party device accepting dial-up connections uses a RADIUS server to check a user identifier and password against a list of valid user identifiers held in the RADIUS server. Typically, this server also allocates a dynamic IP address to the dialup host. The APC can either contain the RADIUS server software itself, or can act as a proxy for RADIUS, i.e. accept RADIUS requests and pass them onto an external RADIUS server. Both configurations allow the APC to perform the following call admission process, which is typical of that for dial-up host access networks:

Monitor the RADIUS authentication protocol. If RADIUS server rejects the user's authentication attempt, call admission fails immediately.

For successful attempts, extract the user identifier and any dynamically-assigned IP address.

Record the (IP address, user identifier) tuple for the host.

On every request or response from this host, extract the IP address and use this to determine the user identifier recorded in step 3. This user identifier is the 'authenticator' for this authentication process.

The APC consults its local policy module (LPM) in order to map the user identifier into a 'policy account' (using a table mapping user identifiers to policy account IDs). If no such policy account exists, the request/response is rejected, otherwise call admission succeeds. The Policy Account tuple, which includes the account name and account ID, is kept by the APC for future use in select QoS 340.

For host access networks that have a variable QoS (e.g. cable modems, analogue modems, etc), the APC will normally measure the actual access network QoS at the first time that a user undergoes the call admission process, and periodically thereafter, just as

with the ISG embodiment. The APC stores this information in the 'measured access network QoS' field its own private copy of the Policy Account for the user.

The host access network QoS is measured with minimal overhead by recording the actual latency for a small number of measurement packets of two differing sizes, S1 and S2, sent from the APC to the client machine, where S1 is the smallest length possible on that network and S2 is the greatest packet length allowed on that network. The details of this measurement process are exactly as for the ISG embodiment.

For host access networks that have fixed QoS (e.g. ISDN), the measured access network QoS field is simply set by the administrator to a standard value based on the capabilities of the host access network.

At the time of authorising a user to use the APC, the corresponding Policy Account's 'QoS limit' is set to a suitable value in the policy service.

Another type of authentication process is used to securely identify the APC system to remote APCs. The purpose of this second authentication stage is to prevent unauthorised use of APC services by client machines masquerading as APCs. To perform this second stage of authentication, the two APCs can go through a standard challenge-response authentication process whenever a new connection is set up between the APCs.

In order to avoid serving users whose APC service subscription has expired, the policy service ensures that the deletion of a Policy Account is reflected within a short time in all policy caches holding a copy of this policy data. As a result, any request or response from a user after their account has expired will be refused by the APC due to failure to find a valid policy account.

Select QoS

This section describes the select QoS stage 340 from fig. 12. As throughout the APC, QoS is defined as a tuple of the form (bandwidth, latency), with one special bandwidth value, -1, indicating that best-effort service should be used.

This stage uses the policy service extensively in order to select the required QoS. The policy service is located on one or more separate machines known as policy servers, forming a hierarchy in which each policy server caches responses from its superior policy server. This provides a highly scalable implementation capable of supporting a world-wide user base.

Each APC is configured to contact its nearest policy server (in the network topology), and maintains its own policy cache based on the responses. In order to process policy requests, the local policy module on the APC (i.e. the client of the policy service) first checks the policy cache, in case the response is already available there, then if necessary passes each request to the policy service. Frequently, new requests and responses arriving at an APC will be processed largely using data from the policy cache. The overall approach is somewhat similar to the ISG embodiment but acts on responses, not just requests, and is more scalable.

When performing said QoS selection stage, the APC will perform the following steps, outlined in fig 14:

The APC analyses the request or response 441 in order to extract 'key elements' that will be used in determining the QoS for this request/response. The exact elements that are used are protocol dependent, so HTTP will be used as an example here, although any IP-based protocol can be handled in this embodiment. For a given HTTP transaction, both the client-side and server-side APCs analyse the HTTP request in order to determine the HTTP version for later use. Then,

once the HTTP server has begun its response, the APC analyses the initial part of the response (i.e. the response headers), using the HTTP version as a guide, to determine two 'key elements', such as the MIME-Type and the Content-Length fields of the header. The former indicates the type of data included in the response, while the latter indicates the length of the file being delivered.

The APC uses the request/response's server port number (i.e. the port number denoting the application), together with the key elements extracted, as parameters in two or more requests to the policy service that together determine 442 the QoS required for this type of request/response within the given protocol. The policy service uses the key elements in these requests to return a final response that indicates the QoS value required. For the example of HTTP, the policy service uses the port number 80 to determine it should use an HTTP-specific mapping table, held in a relational database, of the form (MimeType, MinLength, MaxLength, FlowTypeID), which provides from a given MIME type and content length a unique 'flow type ID'. This unique ID is then used in a second policy request, which maps the FlowTypeID into a QoS value (i.e. bandwidth and latency). Similar mapping tables exist for other protocols that are supported by the APC; if a protocol is unsupported or a matching entry does not exist, fall-back policy data is used (described at the end of this stage), transparently. This QoS value is termed the 'object QoS' since it is determined by the object being delivered by the response.

The APC sends a policy request to the policy service 443, using as parameter the Policy Account ID obtained by call admission 320, in order to obtain the QoS (i.e. bandwidth and latency) allowed by this user's APC service subscription.

The APC calculates the current QoS 444 of the host access network, taking into account any other responses that may be currently proceeding over that access network, either to or from the host whose request/response is being processed. The APC uses the 'measured access network QoS' field (taken from the APC's private copy of the Policy Account for this user) to establish the total bandwidth available, then subtracts, for each response currently proceeding through the APC to the host, the bandwidth element of the QoS selected for that response. The result is called 'available access network bandwidth' and is held by the APC for later use in this QoS selection stage. We assume latency is fixed for the host access network, since the APC is carefully managing the load on the host access network due to this host, and in particular the APC is performing traffic shaping on all responses to avoid overloading the host access network.

Now that the various types of allowed and desired QoS have been established, the APC combines them to determine the selected QoS 445.

First, the APC sets the bandwidth element of this combined QoS to the minimum of its current value and the 'available access network bandwidth' calculated earlier in this stage. The APC then sets the latency element of this combined QoS to the maximum of its current value and the latency element of the 'measured access network QoS' calculated in call admission 320. This 'combined QoS' represents the maximum bandwidth and minimum latency that can be supported by the user's subscription when combined with the host access network currently in use.

Secondly, the APC computes the 'desired QoS' from this 'combined QoS' and the 'object QoS' required by the request/response. This 'desired QoS' refers only to the

QoS that must be provided by the wide area network, not to the QoS provided by other elements such as the host access network or the content server itself.

Specifically, the bandwidth element of the 'desired QoS' is calculated as the minimum of the bandwidth elements of the combined QoS and the object QoS. The latency element of the 'desired QoS' is calculated by subtracting the latency element of the allowed access network QoS from the latency element of the object QoS. This ensures that both the latency possible with the host access network and the latency allowed by the user subscription are accounted for, by reducing the acceptable wide area network latency by a corresponding amount.

The result of this 'desired QoS' calculation may indicate that the object QoS cannot be provided, owing to the user subscription, host access network or current traffic on the host access network. In particular, either a desired QoS bandwidth that is less than the object QoS bandwidth, or a desired QoS latency that is less than or equal to zero, indicate that the desired QoS is impossible, in which case the APC sets the bandwidth element of the QoS to -1, indicating best-effort. The request is then handled via the 'achieve QoS', where the '-1' bandwidth will cause the transaction to use the best-effort network.

When an application protocol is not supported by the APC, or where policy data about a particular key element (e.g. an HTTP MIME type) of an application request/response is not defined in the policy service, it is important that fall-back policy data is always available. This fall-back policy data is employed to attempt to estimate the QoS parameters required by the request/response. Although this policy data is presented here, it is automatically used in step 2 above, which looks up the 'object QoS'; if the key elements used in the policy lookup do

not specify any policy data, the policy service instead returns fall-back policy data, transparently to the APC.

One or more of the following types of fall-back policy data may be employed to derive an estimate of the required QoS - the choice of which fall-back policy data will be used is entirely up to the administrator of the policy service, but port-based policy data is always present to handle unsupported protocols:

Port-based fall-back policy derives the QoS directly from the server-side port number denoting the protocol (i.e. the destination port number for a request, or the source port number for a response). A mapping table of the form (port number, QoS) is used.

host/network-based fall-back policy is useful to assign a default QoS to all traffic to or from a particular host, or a particular host access network or part thereof. Exceptions are still possible using port-based or 'key element'-based policy.

HTTP URL (Uniform Resource Locator) fall-back policy is useful where a wide variety of MIME types may be in use, but some component of the URL can be used to determine the QoS. For example, the URL can supply a directory such as '/newsfeed/' which is known to contain high-priority data that requires high QoS due to its business value, even though its MIME type is identical to low-priority data.

Many other types of policy data are possible, depending on the protocols in use and the applications that use them.

Check Cache

The client-side APC optionally includes a data cache for storing frequently accessed data to facilitate speed and QoS of data retrieval. This data cache may even be external, provided by a

separate data cache server, e.g. an HTTP proxy cache server. Thus, for requests in any protocol where the responses are cacheable (e.g. HTTP in IP networks), the APC checks the data cache for the response to a request before sending a request across a network for data in the form of an object.

When performing the cache checking stage 360, the system uses the name of the object requested as an index into a 'cache table' of (object name, object response, object QoS) tuples. If the said named object exists in the table then the request may be finished by sending the associated object response back to the client using the object QoS from this table, as shown at 370 in fig. 12. This 'cache table' is built up by storing responses to earlier requests as they pass through the ISG, including the object QoS. Note that the object QoS is determined only by the object, and can be cached, whereas the 'desired QoS' actually used when delivering the object to a particular user is dependent on that user and host access network, and cannot be cached (except by the policy cache, which is designed to handle this issue).

The details of the check cache process are largely identical to the ISG embodiment; the sole difference is that the APC check cache stage 360 takes place after the select QoS stage 340, which enables the APC to deliver data from cache using not only the object QoS but also the user-specific 'combined QoS'. The two QoS values are combined exactly as described in select QoS stage 340, resulting in the same 'desired QoS' value as would be obtained if the object had not been in cache. This ensures that the presence of the cache is effectively invisible to the user, and enables the APC service provider to choose whether to invest in larger data caches or more QoS network capacity, depending on the degree to which application traffic makes good use of caching and the relative costs of these options.

However the request is fulfilled, the APC ensures that the transaction is logged, as described below.

Achieve QoS

The achieve QoS stage 380 in fig. 12 is expanded in fig. 15 as a decision tree. The aim of said stage is to come to a sound decision about the most efficient way to achieve the quality of service level settled upon in these preceding stages (i.e. the 'desired QoS' value resulting from select QoS 360). This stage is highly similar to the ISG embodiment but is presented here for completeness.

The achieve QoS stage proceeds as follows:

1. Decide whether the desired QoS level may be achieved using a best effort 461 or QoS 462 network. The APC inspects the bandwidth element of the 'desired QoS': if this is -1, indicating best-effort service, the best-effort network is used and the remaining steps are skipped.

2. If the bandwidth element of the desired QoS is greater than zero, the APC decides whether the desired QoS level can be satisfied in one of the following ways:

   a) entirely through the default forwarder service 463 - this is used where the expected response is short - specifically, where the total latency of the entire expected response is less than or equal to 150% of the time to set up a new connection through the QoS network. This ensures that short request/response transactions are processed quickly without undue network overhead.

   b) by routing the first section of the response through the default forwarder service 463 and at the same time establishing a connection through a QoS network for the remainder of the response 464 - this is used where the response is lengthy (i.e. greater than 150% of the time to set up a new connection through the QoS network) and

the latency part of the desired QoS level is small. A small latency is defined as less than or equal to 150% of the time to set up a new connection through the QoS network.

c) by establishing a connection through a QoS network for the whole response 464 - this is used where the response is lengthy (i.e. greater than 150% of the time to set up a new connection through the QoS network) and the latency part of the desired QoS level is large. A large latency is defined as greater than 150% of the time to set up a new connection through the QoS network.

As well as creating a new QoS network connection, the APC may, if traffic shaping is supported by the 'APC platform', set traffic shaping parameters to correspond to the new QoS network connection. These parameters may be set directly (e.g. using the /dev/cbq interface to the AltQ/CBQ traffic shaping used with FreeBSD UNIX in the first implementation) or indirectly (e.g. by the RSVP protocol software using the above /dev/cbq interface as a result of a new RSVP reservation being established).

Creating a new QoS network connection may cause a network error to be returned to the APC if the required QoS cannot be achieved, due to lack of capacity within one or more network nodes or links between the APC and the destination of the connection. As a result of this and other network errors, the APC notifies the user through an 'alert page', as discussed in the ISG embodiment, which is sent either to the user's web browser using HTTP, or for non-HTTP transactions to a small alerter application program unit on the host.

Initiating the wide area network transaction with the desired QoS

After the achieve QoS stage 380 from fig. 12, a wide area network transaction may be initiated given the condition that the object required cannot be found in the APC's local cache. The wide area

network transaction will take place as decided by the algorithms employed in the achieve QoS stage 380. The response may be cached as it is received. Large files and open-ended multimedia streams are not generally cached. Furthermore, it is possible for a server to defeat caching of a particular object as part of the HTTP protocol.

The APC is aided by two well-known network services, and two APC-associated network services, as shown in fig. 13.

One of said network services used by the APC is the billing and customer care system (BACCS) 401, which is used by the ISG service provider to generate bills for customers. The BACCS may be available to the APC over exactly one of the best effort or QoS networks which are connected to the APC. The BACCS is explained in greater detail in the section "Billing And Customer Care System" below. Although it would be possible to conceive of an APC which is not connected to at least one network that has a BACCS, such an APC would not be able to perform billing and a number of other desirable services which are preferable for the current embodiment of the invention.

The second of said network services required by the APC is the default forwarder service 402. The default forwarder service (often also known simply as the default forwarder) must be available where one or more QoS networks is connected to the APC. In the case where the APC is not connected to any QoS network, a default forwarder is not required. The default forwarder is explained below in more detail in the section "default forwarder service".

One network service which is not well-known and is closely linked to the APC is the policy service 403, previously described. The policy service may be available to the APC over exactly one of the best effort or QoS networks which are connected to the APC.

Another APC-associated network service is the BACCS Gateway 404, described below. This interfaces between the BACCS and the policy service, hiding the implementation details of diverse BACCS implementations behind a single interface.

It is possible to conceive of a single logical or physical computer system which performs more than one of the four preceding services. It is also conceivable that a single logical network service will be performed by more than one physical computer system; this might be particularly beneficial to reduce the average distance covered by transactions between APCs and network services or to allow the service to adequately cope with the volume of transactions made by a large population of APCs.

Network usage and transaction logging

It is important that the APC logs transactions which use precious, scarce or costly resources such as the QoS network or the APC's data cache. To ensure this, the APC logs all transactions where (a) their policy data specified a QoS with bandwidth greater than zero or (b) they were served from the APC data cache. This scheme ensures that all QoS traffic is logged, even if cached, without the excessive overhead of logging best-effort traffic. Transaction logs containing one or more transactions are periodically submitted automatically by the APC to the policy service, which buffers these logs and transmits them to the BACCS Gateway 404.

By logging transactions in the manner outlined above, the embodiment of present invention will allow flexible and precise billing to be done, based upon both the cost associated with the network connection itself and the cost of the large on-line disk store used for the APC's data cache.

BACCS Gateway

The BACCS Gateway 404 is a network service that is responsible for translating the transaction logs submitted periodically by the APCs, via the policy service, into a form required by the APC service provider. The BACCS Gateway also coordinates user accounts between the policy service and the BACCS, copying updates made in the BACCS down to the policy service.

Due to the incorporation of content usage references (e.g. URLs for HTTP-accessible content), the BACCS Gateway will also be able to create content usage counts for content sites and advertising material and deliver these to advertisers, advertising agencies, content sites, and others, similarly to the ISG embodiment.

Default forwarder service

The default forwarder 402 is a network service that is implemented identically to the ISG embodiment, using guaranteed quality network connections to each APC and server that is available on the QoS network. It avoids the expense and latency of setting up calls over QoS networks for short transactions, and may be replicated and hierarchical to cope with large networks. Many existing networks provide default forwarder services: for example, RSVP-enabled IP networks can use normal IP forwarding with priority queuing to provide a default forwarder service, while ATM networks may implement MPOA (Multiprotocol Over ATM) and related functionality, which includes a default forwarder service.

Advertisement delivery

One specific example of the use of the embodiments outlined above is as a medium to deliver targeted advertising with a service level guaranteed to the advertiser. In detail, the ISG or APC will enable advertisers to pay to ensure that adverts are delivered with a specified QoS. Payment from the advertiser will

ensure a guaranteed latency (the time between the user requesting the advert and the advert appearing) and a guaranteed bandwidth (the size of network pipe that delivers the advert). The payment method is similar in nature to freephone ('0800', in the UK, or 'toll-free' in the US) telephone information lines.

When used as an advertising delivery service, the ISG will offer two payment models. Model A will deliver adverts directly from local storage on the ISG box itself Model B will deliver adverts by opening a network connection across a QoS network to the advertiser's server.

Model A will be suitable for widely targeted (i.e. mass market) advertising. It will have a high fixed overhead, but almost negligible cost per hit. Model A corresponds to a push-replicated model. and does not, in fact, require either the ISG or the advertiser to be connected to a QoS-aware network.

Model B will be suitable for narrowly targeted (i.e. niche market) advertising. Costs per hit will be high, but there will be only a small fixed charge. Model B corresponds to the ordinary delivery mechanism of pages over the QoS-aware network, and requires both the ISG and the advertising server to be connected to the same QoS-aware network.

There is some scope for combining models A and B. For example, advertisers may choose to place adverts directly into the ISGs (model A) only in certain geographical regions where hits are expected to be very high. Hits outside these regions will be served from a central server (model B). By providing high bandwidth connections across a QoS-aware network or guaranteed space on the ISG, advertisers will be able to deliver high quality multimedia-rich adverts. The adverts will be limited only by the speed of the user's access network. TV-quality adverts could be delivered this way to most users.

The ISG sees all Internet requests that each user makes. It is therefore in an ideal position to collect total reading information about clients.

There is expected to be a subscriber relationship between readers and ISG owners. Thus total readership tracking and traditional retail sales information may be combined to produce a powerful new all-encompassing targeting service. This service will be an enhancement to traditional retail sales receipts analysis, not a competitor. The reader will not need to be a paying subscriber to the other services offered by the ISG in order to view the guaranteed quality adverts. The ISG provides a central point to collect content usage counts for adverts viewed by each user. These content usage counts can be used to fairly charge advertisers for the advertise service under either model A or model B.

Claims:

1. A method of communicating in a communications network having a first node, a second node and an intermediate node, the method comprising:

sending a request, from the first node, for data stored at the second node in the network;

intercepting the request at the intermediate node;

analysing the request to determine the data transfer parameters appropriate to the request; and

supplying the requested data in the manner specified by those parameters.

2. A method according to claim 1, wherein the request is analysed to extract the identification of the node from which the request is made, and the data is supplied with the parameters appropriate to that node.

3. A method according to claims 1 or 2, wherein the request is analysed to extract the identification of the node to which the request is made, and the data is supplied with the parameters appropriate to that node.

4. A method according to claims 1, 2 or 3, wherein the request is analysed to determine the nature of the data requested, and the data is supplied with the parameters appropriate to the nature of the data.

5. A method according to claim 4, wherein the data is stored as a file, and the nature of the data requested is determined as a result of analysing the file type.

6. A method according to claim 5, wherein the file type is denoted by the file extension.

7. A method according to any preceding claim, the network further comprising a descriptive data store for storing descriptive data, the step of analysing comprising:

retrieving the descriptive data from the descriptive data store; and

analysing the descriptive data to determine the transfer parameters appropriate to the request.

8. A method according to claim 7, wherein the descriptive data comprises nature data describing the nature of the data requested, the step of analysing comprising:

retrieving the nature data from the descriptive data store, and

analysing the nature data to determine the transfer parameters appropriate to the request.

9. A method according to claim 7, wherein the nature data describes the file type of the data requested.

10. A method according to claim 7, wherein the nature data describes the bandwidth required by the data requested.

11. A method according to claim 7, wherein the descriptive data comprises identifiers of the first and second nodes and access information, the step of analysing comprising:

retrieving the identifiers of the first and second nodes and access information from the descriptive data store, and

analysing the identifiers and access information to determine whether the first node is entitled to access data from the second node.

12. A method according to claim 7, wherein the descriptive data comprises identifiers of the users associated with the first node and access information, the step of analysing comprising:

retrieving the identifiers of the first and second nodes and access information from the descriptive data store, and

analysing the identifiers and access information to determine whether the first node is entitled to access data from the second node.

13. A method according to claim 8, wherein a fall back estimation of the transfer parameters is made, if there is no nature data available for the data requested.

14. A method according to claim 13, wherein the transfer parameters are determined by analysing the file type of the data requested.

15. A method according to any preceding claim, wherein the transfer parameters comprise the bandwidth of the data transfer.

16. A method according to any preceding claim, wherein the transfer parameters comprise the latency of the data transfer.

17. A method according to any preceding claim, wherein the transfer parameters comprise the jitter of the data transfer.

18. A communications network having a first node, a second node and an intermediate node, and further comprising:

means for sending a request, from the first node, for data stored at a second node in the network;

means for intercepting the request at the intermediate node;

means for analysing the request to determine the data transfer parameters appropriate to the request; and

means for supplying the requested data in the manner specified by those parameters.

19. A network according to claim 18, wherein the means for analysing the request comprises means for extracting the identification of the node from which the request is made.

20. A network according to claims 18 or 19, wherein means for analysing the request comprises means for extracting the identification of the node to which the request is made.

21. A network according to claims 18, 19 or 20, wherein means for analysing the request comprises means for determining the nature of the data requested, and the data is supplied with the parameters appropriate to the nature of the data.

22. A network according to claim 18, wherein the data is stored as a file, and means for analysing the request comprises means for determining the file type.

23. A network according to claim 22, wherein the file type is denoted by the file extension.

24. A network according to any preceding claim, the network further comprising a descriptive data store for storing descriptive data describing aspects of data stored at nodes in the network, the means for analysing further comprising:

means for retrieving the descriptive data describing the requested data from the descriptive data store; and

means for analysing the descriptive data to determine the transfer parameters appropriate to the request.

25. A network according to claim 24, wherein the descriptive data comprises nature data describing the nature of the data requested.

26. A network according to claim 25, wherein the nature data describes the file type of the data requested.

27. A network according to claim 24, wherein the nature data describes the bandwidth required by the data requested.

28. A network according to claim 24, wherein the descriptive data comprises identifiers of the first and second nodes and access information.

29. A network according to claim 24, wherein the descriptive data comprises identifiers of the users associated with the first node and access information.

30. A network according to any preceding claim, wherein the transfer parameters comprise the bandwidth of the data transfer.

31. A network according to any preceding claim, wherein the transfer parameters comprise the latency of the data transfer.

32. A network according to any preceding claim, wherein the transfer parameters comprise the jitter of the data transfer.

33. A system for providing data communications at a specified level of service in communications network, the network having a first node and a second node,
the network comprising:
means for sending a request, from the first node, for data stored at a second node in the network;
the system comprising:
means for intercepting the request;
means for analysing the request to determine the data transfer parameters appropriate to the request; and
means for supplying the requested data in the manner specified by those parameters.

34. A system according to claim 33, wherein the means for analysing the request comprises first means associated with the first node for determining the transfer parameters in relation to the user issuing the request.

35. A system according to claim 33, wherein the means for analysing the request comprises second means associated with the

second node for determining the transfer parameters in relation to the data requested.

36. A system according to claim 35, wherein the second means comprises means for determining the nature of the data requested, and the data is supplied with the parameters appropriate to the nature of the data.

37. A system according claim 35, wherein the second means comprises:

a descriptive data store for storing descriptive data describing aspects of data stored at the second node in the system,

means for retrieving the descriptive data describing the requested data from the descriptive data store; and

means for analysing the descriptive data to determine the transfer parameters appropriate to the request.

38. A system according to claim 37, wherein the descriptive data comprises nature data describing the nature of the data requested.

39. A system according to claim 38, wherein the nature data describes the file type of the data requested.

40. A system according to claim 38, wherein the nature data describes the bandwidth required by the data requested.

41. A system according to claim 34, wherein first means comprises means for analysing identifiers of the users associated with the first node and access information.

42. A system according to any of claims 33 to 41, wherein the transfer parameters comprise the bandwidth of the data transfer.

43. A system according to any of claims 33 to 41, wherein the transfer parameters comprise the latency of the data transfer.

44. A system according to any of claims 33 to 41, wherein the transfer parameters comprise the jitter of the data transfer.

45. A method according to claim 1, further comprising:
intercepting the response at the intermediate node;
analysing the response to determine the data transfer parameters appropriate to the request; and
supplying the requested data in the manner specified by those parameters.

46. A method according to claim 45, wherein the response is analysed to determine the nature of the data requested.

47. A method according to claim 46, wherein the response is analysed to determine the quantity of data requested.

46. A method of communicating in a communications network having a first node, a second node and an intermediate node, the method comprising:
sending a request, from the first node, for data stored at the second node in the network;
intercepting the response at the intermediate node;
analysing the response to determine the data transfer parameters appropriate to the request; and
supplying the requested data in the manner specified by those parameters.

47. A method according to claim 45, wherein the response is analysed to determine the nature of the data requested.

48. A method according to claim 46, wherein the response is analysed to determine the quantity of data requested.

49. A system for providing data communications at a specified level of service in communications network, the network having a first node and a second node,
the network comprising:
means for sending a request, from the first node, for data stored at a second node in the network;
the system comprising:
means for intercepting the response;
means for analysing the response to determine the data transfer parameters appropriate to the request; and
means for supplying the requested data in the manner specified by those parameters.

50. A method according to claim 45, wherein the means for analysing the response comprises means to determine the nature of the data requested.

51. A method according to claim 46, wherein the means for analysing the response comprises means to determine the quantity of data requested.

52. A computer readable storage medium for use in a data processing system, said medium holding program code, wherein the program code is arranged so that:
when the program code is executed by an intermediate computer node in a computer system comprising a first node, a second node and an intermediate node, the intermediate computer node is caused:
to intercept a request sent from a first node to a second node,
to analyse the request to determine the data transfer parameters appropriate to the request, and
to supply the requested data in the manner specified by those parameters.

53. A computer readable storage medium for use in a data processing system, said medium holding program code, wherein the program code is arranged so that:

when the program code is executed by an intermediate computer node in a computer system comprising a first node, a second node and an intermediate node, the intermediate computer node is caused:

to intercept a request sent from a first node to a second node,

and the program code is further so arranged in relation to said first node:

to determine the transfer parameters in relation to the user issuing the request,

to analyse the request to determine the data transfer parameters appropriate to the request in relation to the user issuing the request, and

to supply the requested data in the manner specified by those parameters.

54. A computer readable storage medium for use in a data processing system, said medium holding program code, wherein the program code is arranged so that:

when the program code is executed by an intermediate computer node in a computer system comprising a first node, a second node and an intermediate node, the intermediate computer node is caused:

to intercept a request sent from a first node to a second node,

and the program code is further so arranged in relation to said second node:

to analyse the request to determine the data transfer parameters appropriate to the request in relation to the data requested, and

to supply the requested data in the manner specified by those parameters.

55. A computer readable storage medium for use in a data processing system, said medium holding program code, wherein the program code is arranged so that:

when the program code is executed by an intermediate computer node in a computer system comprising a first node, a second node

and an intermediate node, the intermediate computer node is caused:

to intercept a request sent from a first node to a second node,

and the program code is further so arranged in relation to said second node:

to analyse the request to determine the data transfer parameters appropriate to the request in relation to the nature of the data requested, and

to supply the requested data in the manner specified by those parameters.

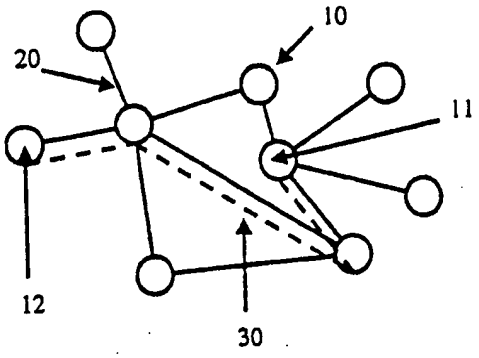


Fig. 1

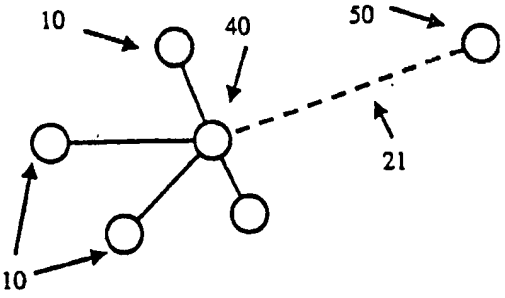


Fig. 2

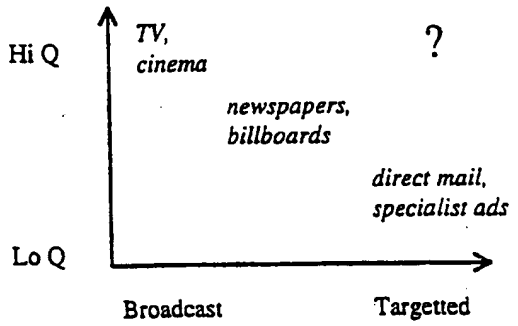


Fig. 3

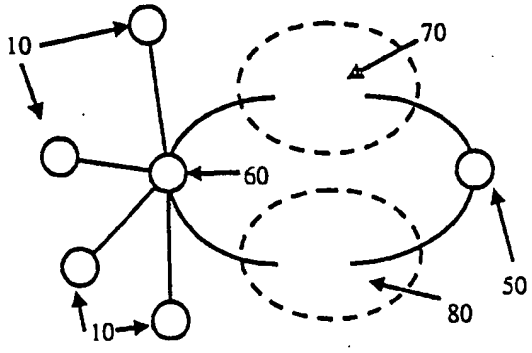


Fig. 4

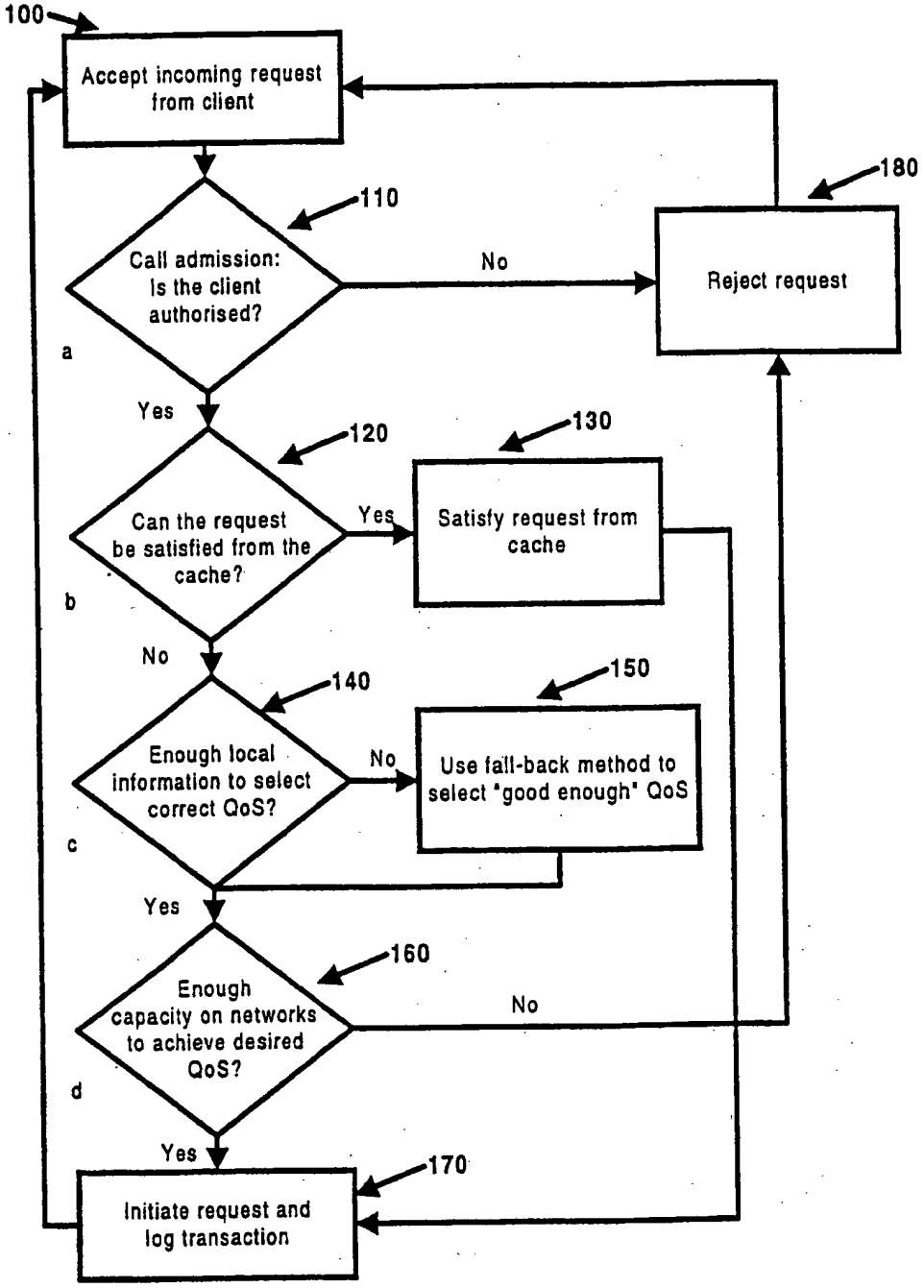

100
Accept incoming request from client
110
Call admission: Is the client authorised?
No
180
Reject request
a
Yes
120
Can the request be satisfied from the cache?
Yes
130
Satisfy request from cache
b
No
140
Enough local information to select correct QoS?
No
150
Use fall-back method to select "good enough" QoS
c
Yes
160
Enough capacity on networks to achieve desired QoS?
No
d
Yes
170
Initiate request and log transaction

Fig. 5

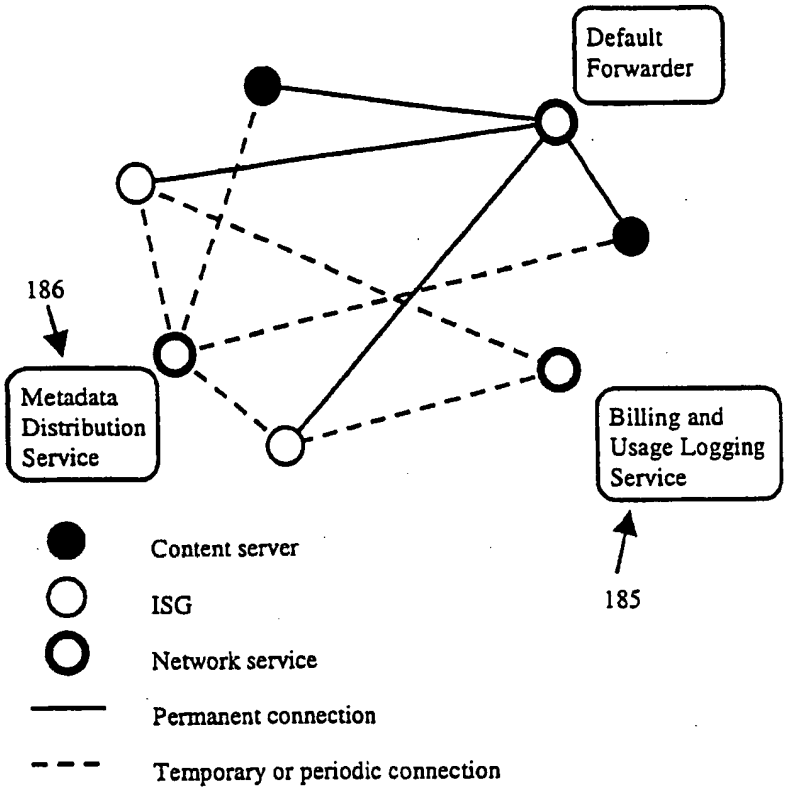

Default Forwarder
186
Metadata Distribution Service
Billing and Usage Logging Service
185
Content server
ISG
Network service
Permanent connection
Temporary or periodic connection

Fig. 6

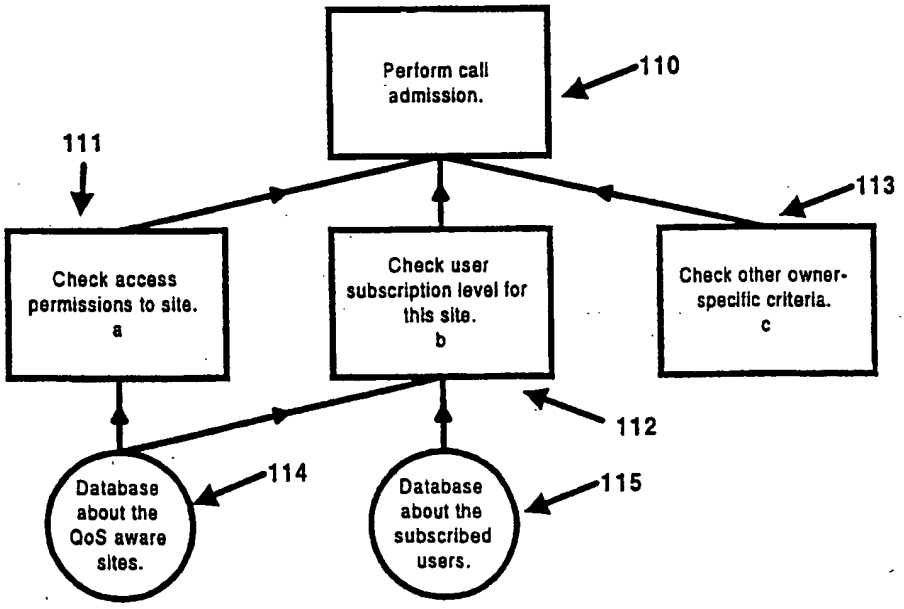

Perform call admission.
110
111
113
Check access permissions to site.
a
Check user subscription level for this site.
b
Check other owner-specific criteria.
c
112
Database about the QoS aware sites.
114
Database about the subscribed users.
115

Fig. 7

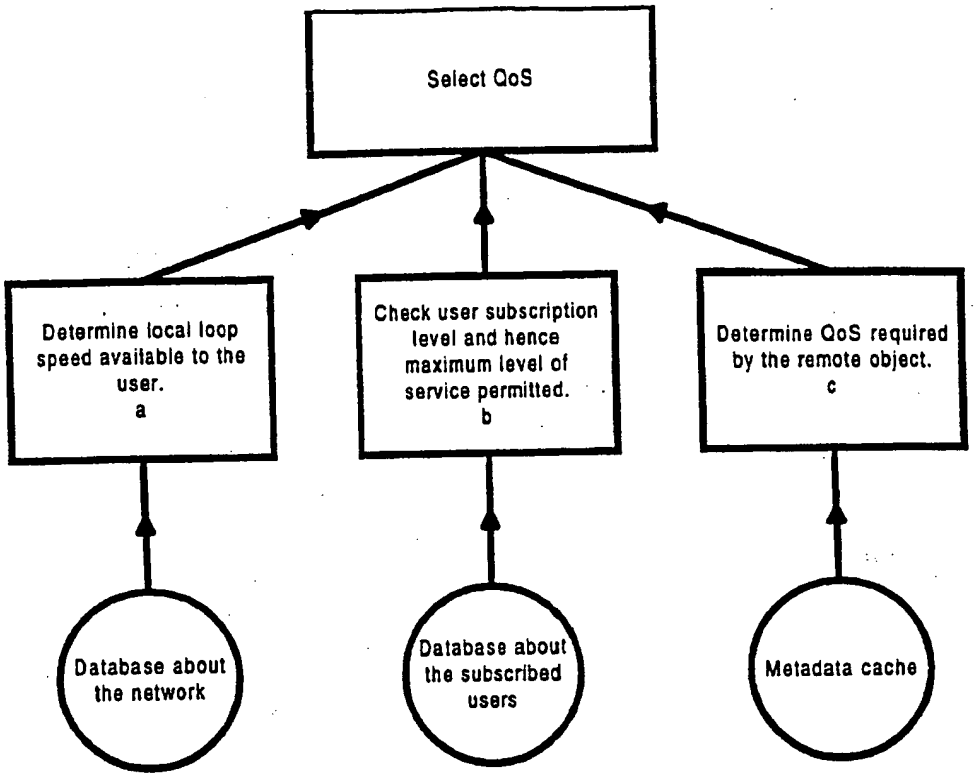

Select QoS
Determine local loop speed available to the user.
a
Check user subscription level and hence maximum level of service permitted.
b
Determine QoS required by the remote object.
c
Database about the network
Database about the subscribed users
Metadata cache

Fig. 8

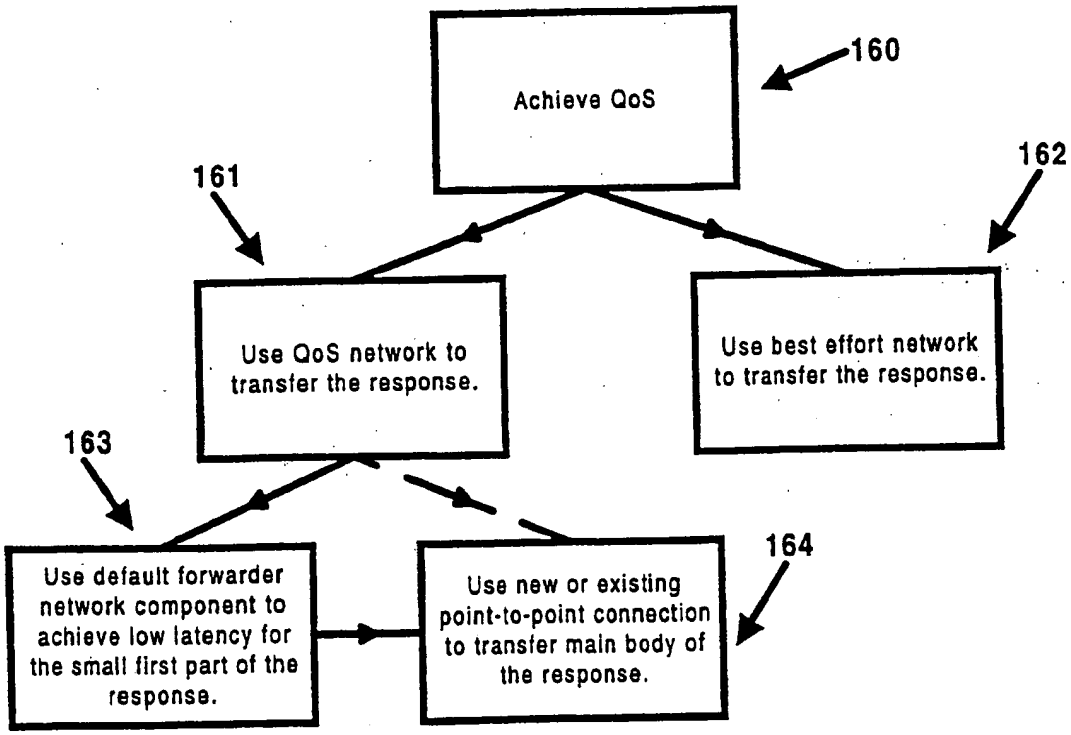

Achieve QoS
160
161
162
Use QoS network to transfer the response.
Use best effort network to transfer the response.
163
Use default forwarder network component to achieve low latency for the small first part of the response.
Use new or existing point-to-point connection to transfer main body of the response.
164

Fig. 9

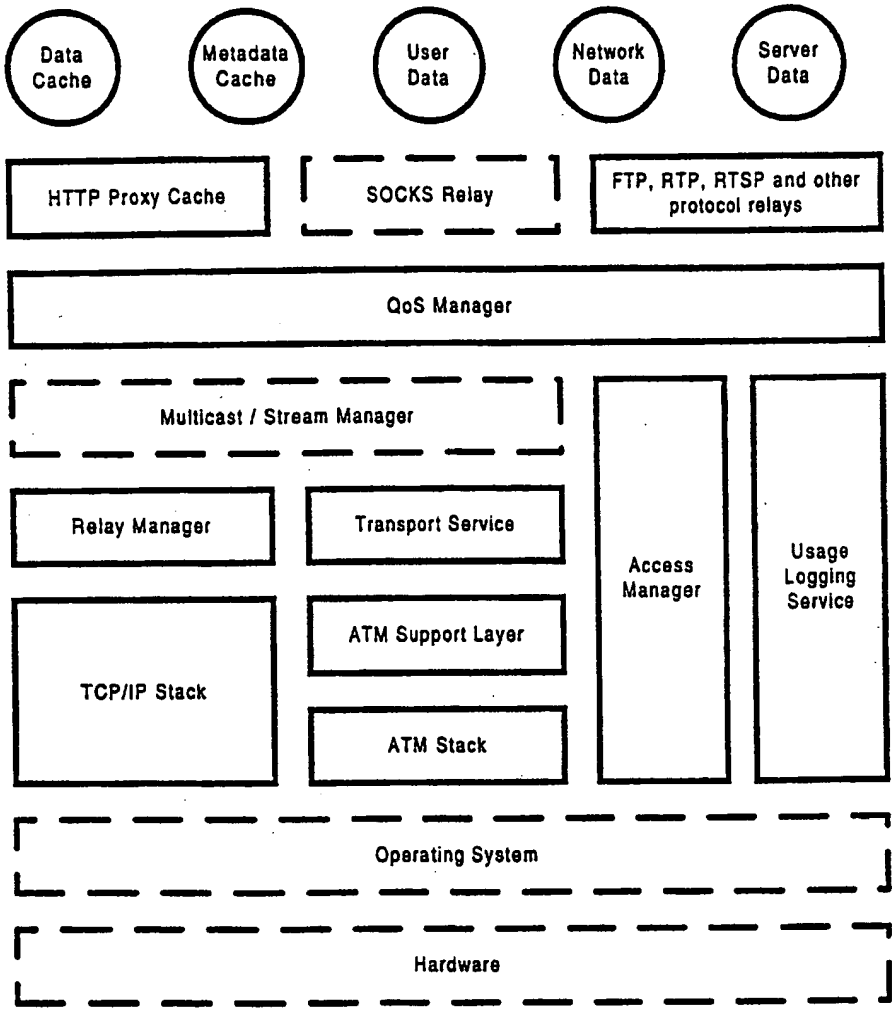

Data Cache
Metadata Cache
User Data
Network Data
Server Data
HTTP Proxy Cache
SOCKS Relay
FTP, RTP, RTSP and other protocol relays
QoS Manager
Multicast / Stream Manager
Relay Manager
Transport Service
Access Manager
Usage Logging Service
TCP/IP Stack
ATM Support Layer
ATM Stack
Operating System
Hardware

Fig. 10

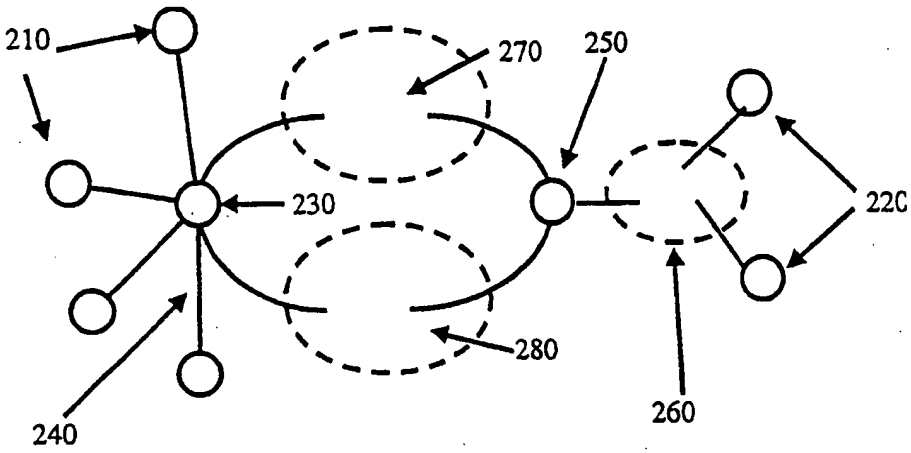


Fig. 11

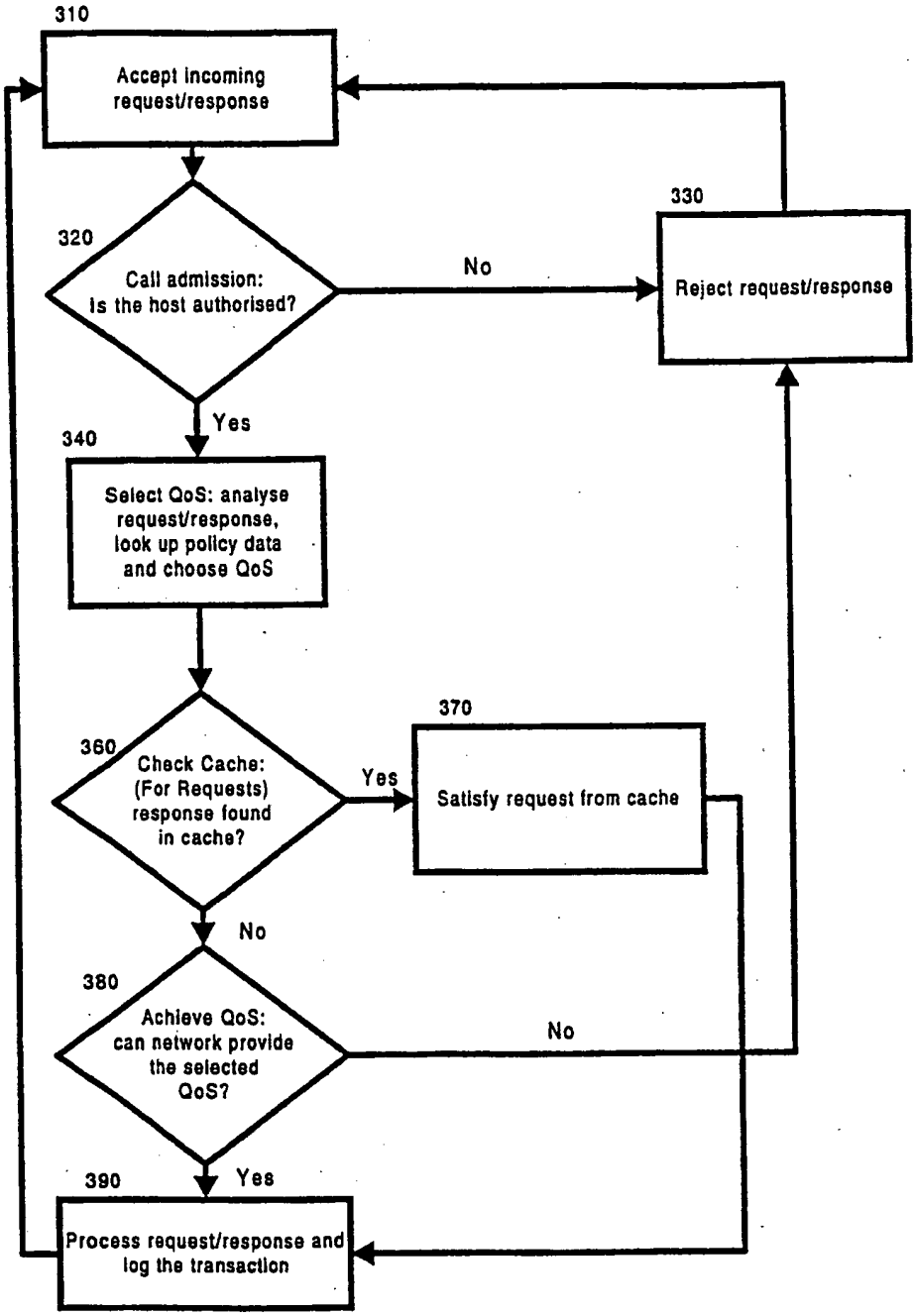

310
Accept incoming request/response
320
Call admission: Is the host authorised?
No
330
Reject request/response
Yes
340
Select QoS: analyse request/response, look up policy data and choose QoS
360
Check Cache: (For Requests) response found in cache?
Yes
370
Satisfy request from cache
No
380
Achieve QoS: can network provide the selected QoS?
No
390
Yes
Process request/response and log the transaction

Fig. 12

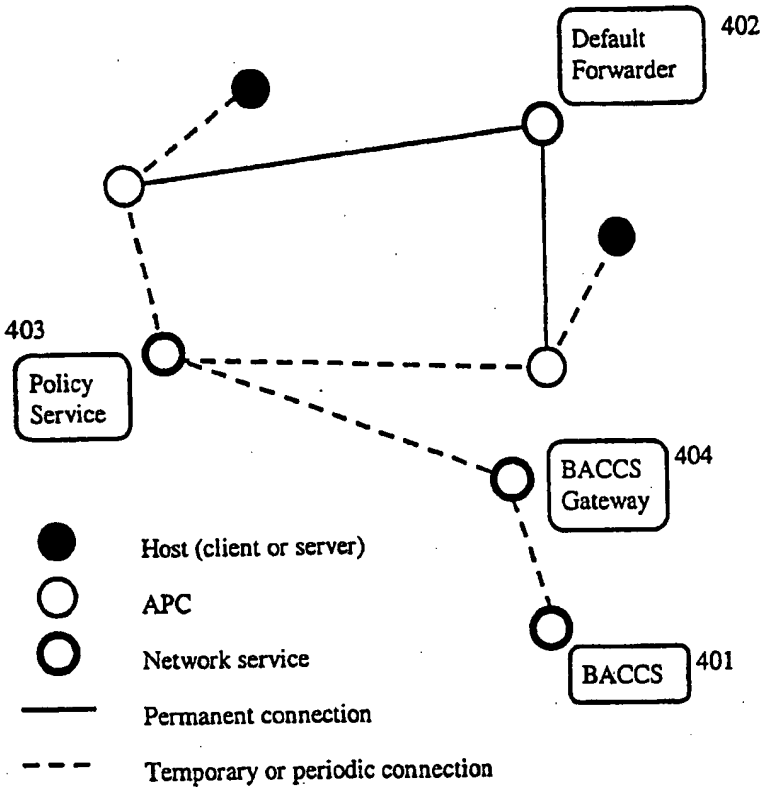

Default Forwarder
402
403
Policy Service
BACCS Gateway
404
BACCS
401
Host (client or server)
APC
Network service
Permanent connection
Temporary or periodic connection

Fig. 13

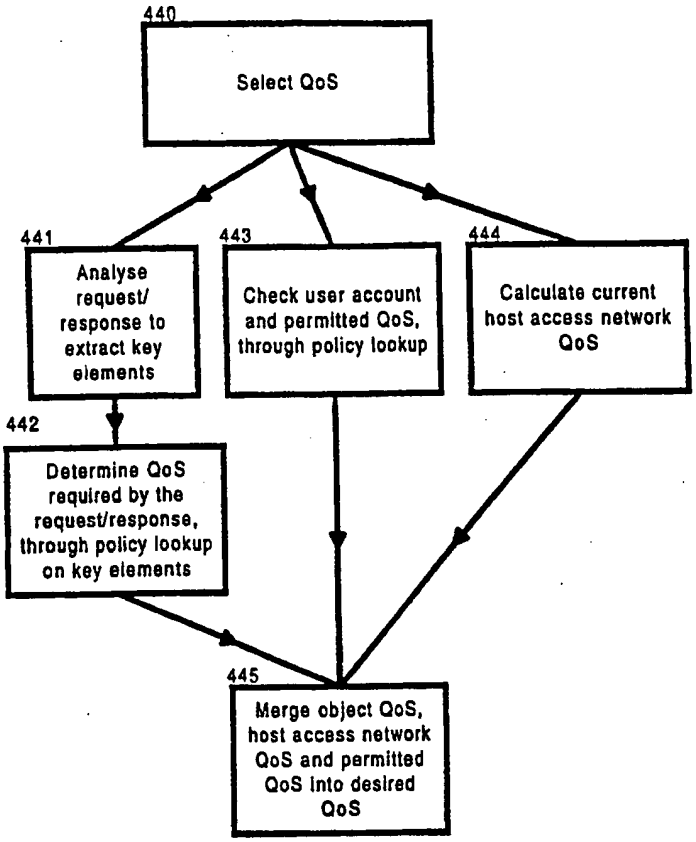

440
Select QoS
441
Analyse request/ response to extract key elements
443
Check user account and permitted QoS, through policy lookup
444
Calculate current host access network QoS
442
Determine QoS required by the request/response, through policy lookup on key elements
445
Merge object QoS, host access network QoS and permitted QoS into desired QoS

Fig. 14

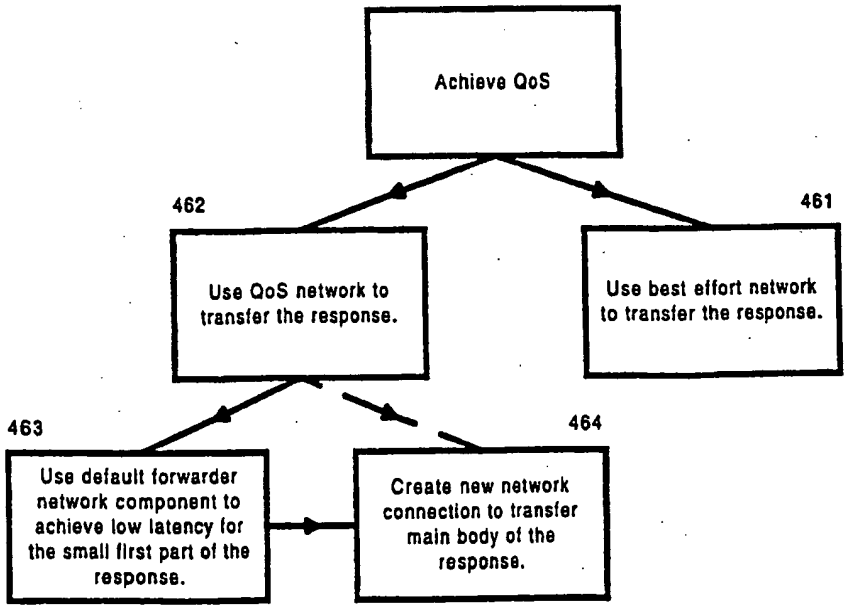

Achieve QoS
462
Use QoS network to transfer the response.
461
Use best effort network to transfer the response.
463
Use default forwarder network component to achieve low latency for the small first part of the response.
464
Create new network connection to transfer main body of the response.

Fig. 15

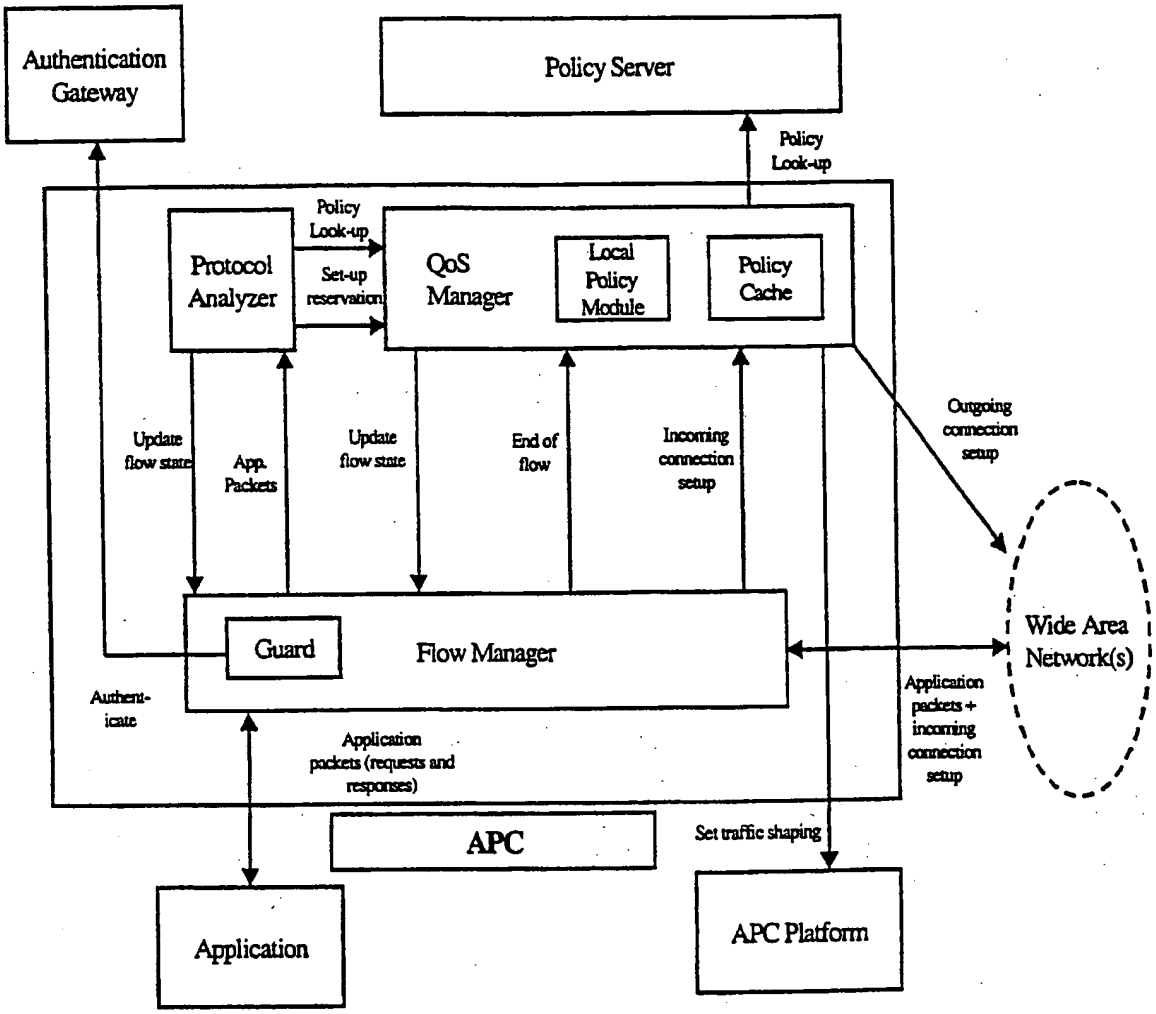

Authentication Gateway
Policy Server
Policy Look-up
Policy Look-up
Protocol Analyzer
Set-up reservation
QoS Manager
Local Policy Module
Policy Cache
Update flow state
App. Packets
Update flow state
End of flow
Incoming connection setup
Outgoing connection setup
Guard
Flow Manager
Wide Area Network(s)
Authent-icate
Application packets (requests and responses)
Application packets + incoming connection setup
APC
Set traffic shaping
Application
APC Platform

Fig. 16

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

This Page Blank (uspto)